



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TRABAJO FINAL DE GRADO

Kamal Khdili, Khalid

ESTUDIO DE LAS ETAPAS DE AUTOMATIZACIÓN DE UNA APLICACIÓN INDUSTRIAL BASADA EN LA GESTIÓN Y TRANSPORTE DE PAQUETES



Director del TFG: Delgado Prieto, Miguel
Convocatoria: Junio, 2020

SUMARIO

1	Introducción	7
1.1	Abstract	7
1.2	Objetivo	9
1.3	Alcance	9
1.4	Requerimientos	9
1.5	Justificación	10
2	Consideraciones previas.....	11
2.1	Normativa.....	11
2.2	Controladores lógicos programables (PLC)	11
2.2.1	Definición	11
2.2.2	Estructura	12
2.2.3	¿Por qué se utilizan los Controladores lógicos programables?.....	12
2.2.4	Ciclo de funcionamiento de un PLC.....	13
2.3	Protocolos de comunicación	14
2.3.1	Protocolo Profibus.....	14
2.3.2	Protocolo CAN	14
2.3.3	Ethernet.....	14
2.4	Software de programación.....	15
2.4.1	Lenguajes de programación	15
2.4.2	Sección de programación	18
2.4.3	Tabla de animación	19
2.4.4	Pantallas de operador	20
2.4.5	Tipos de datos y variables	21
2.4.6	Biblioteca de funciones	23
2.4.7	Tipos de direcciones	24
2.4.8	Bit de sistema	24
3	Celda de trabajo	25
3.1	Introducción	25
3.2	Componentes de la celda	26
4	Tecnologías de identificación	33
4.1	Introducción	33
4.2	Elección de la tecnología de reconocimiento (caso de estudio)	34
4.3	Dispositivo RFID.....	36
4.3.1	Definición	36
4.3.2	Estructura RFID.....	36

Figura 11 - Apartado Sección	18
Figura 12 - Crear una nueva Sección	18
Figura 13 - Orden de las Secciones	19
Figura 14 - Crear Tabla de animación.....	19
Figura 15 - Ejemplo real Tabla de animación	20
Figura 16 - Ejemplo real Tabla de animación	20
Figura 17 - Ejemplo Pantallas de operador	20
Figura 18 - Crear Nueva Pantalla operador.....	21
Figura 19 - Ejemplo variable Alocatada y no Alocatada.....	21
Figura 20 - Variables elementales	22
Figura 21 - Crear Derived Data Type	22
Figura 22 - Representación lenguaje ST	23
Figura 23 - Representación lenguaje FBD	23
Figura 24 - Celda Industrial	25
Figura 25 - Distribución de los protocolos.....	25
Figura 26 - Sensor inductivo.....	26
Figura 28 - Sensor inductivo basculante, no detecta llegada bandeja.....	27
Figura 28 - Sensor inductivo basculante, detecta llegada bandeja.....	27
Figura 29 - Sensor inductivo retenedor y pieza metálica detectada.....	27
Figura 30 - Sensor capacitivo.....	28
Figura 31 - Sensores fotoeléctricos de fibra óptica.....	28
Figura 32 - Electroválvula o válvula solenoide	29
Figura 33 - Plataforma en estado de bajada, de reposo y subida	29
Figura 34 - Retenedor.....	30
Figura 35 - Pulsador.....	30
Figura 36 - Seta de emergencia.....	31
Figura 37 - Motor BOSCH	31
Figura 38 - Componentes RFID.....	36
Figura 39 - Ejemplo lector RFID corte alcance.....	36
Figura 40 - Ejemplo Lector RFID	36
Figura 41 - Estructura de Etiqueta RFID	37
Figura 42 - Funcionamiento RFID	38
Figura 43 - Modelo Industria 3.0/ Modelo Industria 4.0.....	38
Figura 44 - Direcciones de movimiento de la celda	40
Figura 45 - Recorrido de los paquetes	41
Figura 46 - Diagrama de comunicación.....	42
Figura 47 - Comunicación entre las líneas.....	42
Figura 48 - Posición de los sensores.....	43
Figura 49 - Combinaciones Plataformas/Retenedores	44
Figura 50 - Inicialización Variables	45
Figura 51 - Sección SETANDRESET.....	45
Figura 52 - Línea Ladder activación Ready	45
Figura 53 - Línea Ladder activación Move.....	46
Figura 54 - Direccionamiento del sensor	46
Figura 55 - Línea CAN y Profibus	46

4.3.3	Opciones de programa	37
4.3.4	Funcionamiento RFID	38
4.3.5	Sensor RFID en la Industria 4.0.....	38
5	Diseño del proceso de automatización	40
5.1	Aplicación	40
5.2	Comunicaciones industriales.....	42
5.3	Posición de los sensores de reconocimiento	43
5.4	Retenedores y plataformas	44
5.5	Sensores RFID	50
5.5.1	Activación y lectura	50
5.5.2	Identificación del paquete y tratamiento del dígito	53
5.5.3	RFID 1,2,3	57
5.5.4	RFID 4,5,6	58
5.6	Motores	60
5.7	Trazabilidad	61
5.8	Estaciones de trabajo	63
6	Validación	68
7	Conclusiones.....	73
8	Bibliografía	74

SUMARIO - TABLAS

Tabla 1 – Direcciones	24
Tabla 2 - Tecnología de identificación (Criterios).....	34
Tabla 3 - Clasificación de los dígitos.....	53

SUMARIO FIGURAS

Figura 1 - Plano Celda de Trabajo.....	7
Figura 2 - Work cell plan.....	8
Figura 3 - Flujo funcionamiento del PLC.....	11
Figura 4 - Ejemplo PLC.....	12
Figura 5 - Ciclo de SCAN	13
Figura 6 - Diagrama de Contactos (Ladder).....	15
Figura 7 - Ejemplo FBD	16
Figura 8 - Ejemplo Lista de Instrucciones	16
Figura 9 - Texto estructurado – ST	17
Figura 10 - Ejemplo de Etapas Grafset	17

Figura 56 - Sección comunicación PT08/PT15.....	47
Figura 57 - Intersección 1 entrada/2 salidas	47
Figura 58 - Línea Ladder Activación Move 2 direcciones	47
Figura 59 - Intersección 2 entrada/1 salidas	48
Figura 60 - Línea Ladder activación Move intersección 2 entrada/1 salidas	48
Figura 61 - - Línea Ladder prioridad	48
Figura 62 - Línea Ladder desactivación Move sin sensor retenedor.....	49
Figura 63 - Línea Ladder desactivación Move con sensor retenedor	49
Figura 64 - Línea Ladder salida retenedor.....	49
Figura 65 - Sección comunicación DIR06	50
Figura 66 - Línea Ladder simulación lectura RFID	50
Figura 67 - Pantalla operador Introducción/Activación RFID.....	51
Figura 68 - Ejemplo Campo de entrada (escrituraRFID01)	51
Figura 69 - Ejemplo introducción de dígito sensor 1.....	52
Figura 70 - Ejemplo Tabla de animación (escrituraRFID01)	52
Figura 71 - Ejemplo simulación de lectura (escrituraRFID01)	52
Figura 72 - Ejemplo dígitos.....	53
Figura 73 - Ejemplo tratamientos del último dígito	54
Figura 74 - Último dígito.....	54
Figura 75 - Ejemplo operación AND (Último dígito).....	54
Figura 76 - Penúltimos dígito	55
Figura 77 - Ejemplo tratamientos del penúltimo dígito.....	55
Figura 78 - Ejemplo operación AND (penúltimo dígito)	55
Figura 79 - Desplazar 4 bits	56
Figura 80 - Antepenúltimos dígito.....	56
Figura 81 - Ejemplo tratamientos del antepenúltimo dígito.....	56
Figura 82 - Ejemplo operación AND (antepenúltimo) dígito)	57
Figura 83 - Desplazar 8 bits	57
Figura 84 - Línea Ladder selección de producto	57
Figura 85 - Sección comunicación registros	58
Figura 86 - - Línea Ladder activación M102	58
Figura 87 - Línea Ladder asignación de registros	58
Figura 88 - Línea ST activación (PT16_DIRECCION_DIR07/PT16_DIRECCION_DIR017)	59
Figura 89 - Línea Ladder activación VAR_PRODUCTO_FIN/Reset (M100, M200, M300)	59
Figura 90 - Línea Ladder Motores	60
Figura 91 - Posición Motores.....	60
Figura 92 - Pantalla del operador ID	61
Figura 93 - Array Tiempo.....	61
Figura 94 - Pantalla del operador (tiempos de trabajo).....	62
Figura 95 - Tabla de animación (registros).....	62
Figura 96 - Pantalla del operador (registros)	62
Figura 97 - Plano Celda de trabajo (Estaciones de trabajo)	63
Figura 98 - Línea Ladder (estación de carga).....	63
Figura 99 - Línea Ladder (estación de carga) temporizador TON	63
Figura 100 - Línea Ladder Bloque PT06.....	64

Figura 101 - Línea ST (Existencia materias primas)	64
Figura 102 - Línea ST (Disminuir materias primas).....	64
Figura 103 - Línea Ladder activación (PT09_DIRECCION_DIR14/PT09_DIRECCION_DIR11).....	65
Figura 104 - Línea ST activación (TRABAJO_ESTACION_1).....	65
Figura 105 - Línea Ladder activación (M102)	65
Figura 106 - Línea Ladder activación y temporización (estación 1)	66
Figura 107 - Línea ST (TIEMPO_E_1)	66
Figura 108 - Línea ST (Estación de trabajo 1)	66
Figura 109 - Línea ST activación (BANDEJA_CON_PRODUCTO)	67
Figura 110 - Línea Ladder activación (VAR_PRODUCTO_FIN)	67
Figura 111 - Línea Ladder activación y temporización (estación de extracción)	67
Figura 112 - Línea Ladder activación (SUBE_PT05)	68
Figura 113 - Línea Ladder activación (M9)	68
Figura 114 - Pantalla del operador (RFID01, RFID04).....	69
Figura 115 - Ejemplo Tabla animación (escrituraRFID01)	69
Figura 116 - Línea Ladder simulación (Estación de trabajo)	69
Figura 118 - Pantallas de operador (visualización tiempo real)	70
Figura 117 - Tabla animación (tiempo de trabajo).....	70
Figura 119 - Pantallas de operador introducción (RFID01)	70
Figura 120 - Línea Ladder activación (Producto2).....	70
Figura 121 - Línea Ladder activación dirección (PT09_DIRECCION_DIR11)	71
Figura 122 - Línea Ladder acción correctora (Temporizador TON).....	71
Figura 123 - Pantallas de operador (ID sensores, Tiempos de trabajo)	72
Figura 125 - Tabla de animación (ARRAY_TIEMPO_P_F/ARRAY_TIEMPO_M_P)	72
Figura 124 - Pantallas operador (ID sensores)	72

1 Introducció

1.1 Abstract

En este proyecto se ha realizado la simulación de un proceso industrial de paquetes en el laboratorio de la Universidad Politécnica de Cataluña, Terrassa.

El proceso industrial está formado por cinco estaciones de trabajo donde se ubicaría un robot, seis sensores de reconocimiento y una estación de entrada para los paquetes vacíos. Todos estos elementos mencionados serán simulados.

El proceso consiste en llenar los paquetes con el producto correspondiente y dirigirlos a las estaciones de trabajo que les corresponden.

En esta línea hay tres tipos de producto que se introducirán en los paquetes vacíos. Las cajas vacías ya tendrán determinado que tipo de producto debe ir en ellas gracias a los identificadores. Además, la línea gestiona en todo momento donde está situado cada paquete y también determina la dirección a la que debe ir para su acabado.

Este proyecto consta de la programación del PLC para llevar a cabo las acciones simuladas.

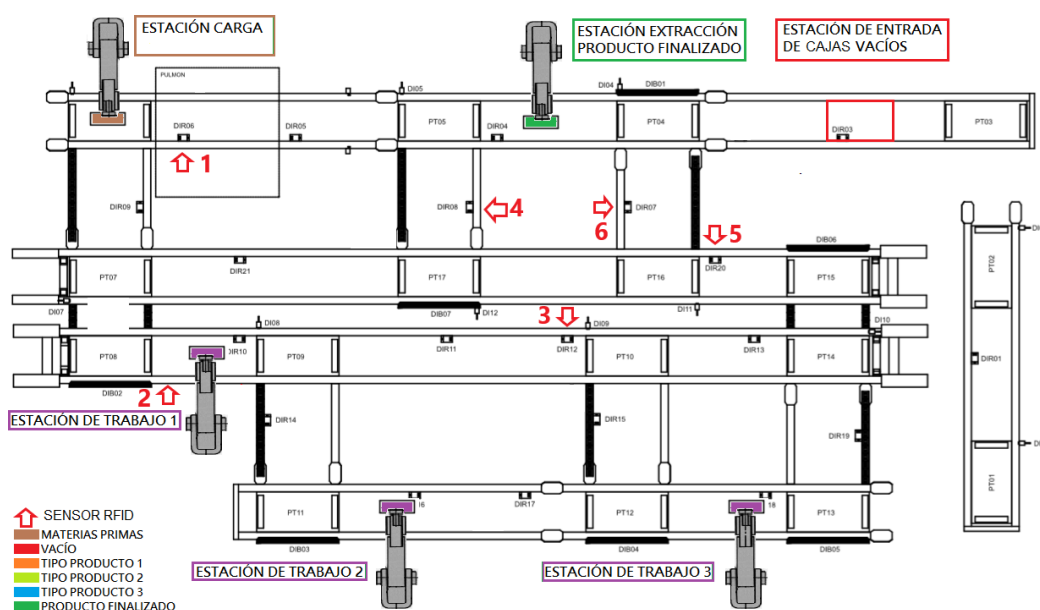


Figura 1 - Plano Celda de Trabajo

In this project, the simulation of an industrial package process has been carried out in the laboratory of the Polytechnic University of Catalonia, Terrassa.

The industrial process consists of five workstations where a robot would be located, six recognition sensors and an entry station for empty packages. All these elements will be simulated.

The process consists of filling the packages with the corresponding product and directing them to the corresponding workstations.

In this line there are three types of product that will be introduced in empty packages. The empty boxes will already have determined which kind of product should go in them thanks to the identifiers. In addition, the line manages at all times where each package is located and also determines the direction to go in for its finishing.

This project consists of the PLC's programming to carry out the simulated actions.

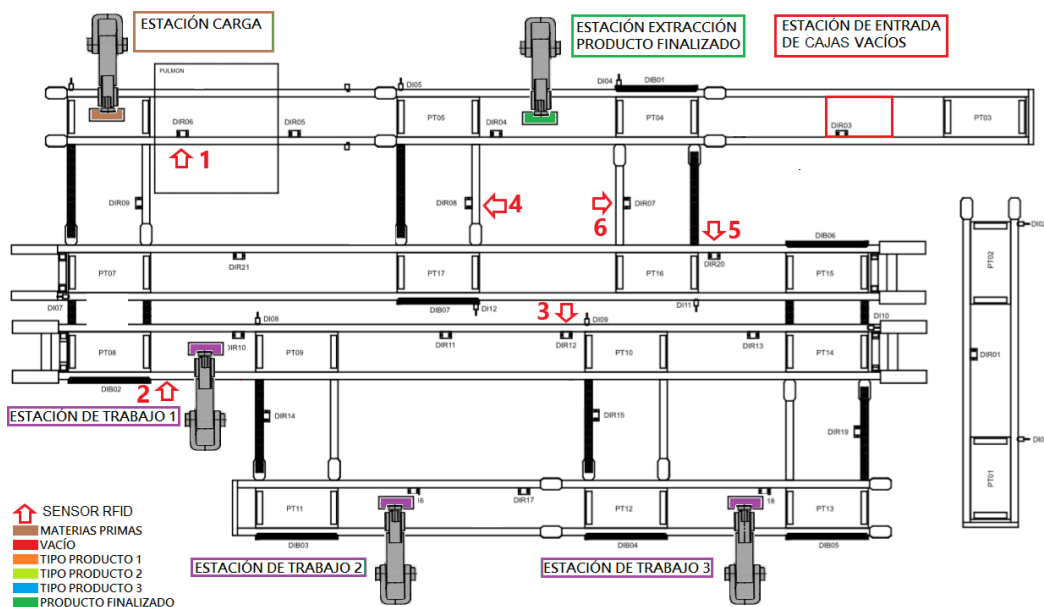


Figura 2 - Work cell plan

1.2 Objetivo

El objetivo principal de este proyecto es gestionar la distribución de paquetes mediante una aplicación industrial basada en la automatización. Para lograr la correcta gestión se estudiará las distintas etapas de automatización.

Además, se realizará un estudio de las diferentes opciones de identificación de objetos que hay en el mercado con la finalidad de escoger la opción que se adapte mejor a nuestro sistema.

Para lograr dicho objetivo se procederá a seleccionar la tecnología de identificación, gestionar las distintas comunicaciones entre los PLCs de la celda de trabajo, diseñar el sistema de automatización y a la validación del sistema.

1.3 Alcance

En este proyecto se realizarán una serie de tareas para lograr los objetivos establecidos. En primer lugar, es necesario buscar e informarse del funcionamiento de la línea de producción. Posteriormente determinar cuál es la manera más óptima y eficiente de llevar a cabo nuestros objetivos.

Comprender la celda de trabajo, entender el conjunto de sus elementos, las comunicaciones que la forman y su funcionamiento. Una vez observado todos estos aspectos procederemos a realizar las siguientes tareas:

- Analizar los diferentes métodos de reconocimiento disponibles actualmente en el mercado, y posteriormente seleccionar la opción que más se adapte a nuestras necesidades.
- Revisar y entender los buses comunicaciones entre los PLCs de nuestra celda. Definir la arquitectura y el flujo de la información.
- Determinar la mejor posición para nuestros sensores de reconociendo dentro la de celda de trabajo.
- Definir la arquitectura de programación, programar de forma que se logre una gestión óptima de nuestro sistema.
- Realizar un presupuesto general de nuestro proyecto y un pliego de condiciones.

La celda de trabajo del laboratorio ya estaba en funcionamiento, contenía un programa base que será el punto de partida del proyecto.

1.4 Requerimientos

Existen una serie de especificaciones que se deben tener en cuenta en el proyecto. A continuación, se numerará las especificaciones de nuestro proyecto:

- Un paquete por estación.
- Limitación de paquetes vacíos, la estación solo llevara el registro de tres paquetes vacío a la vez en la celda de trabajo. Se ha limitado en este número ya que creemos que es más que suficiente y no debería haber más de tres paquetes vacíos a la vez en la celda.
- Distancian entre paquete y sensor.
- Paquete situado siempre encima de la bandeja.

1.5 Justificación

Como estudiante de ingeniería electrónica industrial y automática siempre he mostrado más interés por la automática. Desde un inicio tenía claro que mi proyecto iba a ir enfocado a la automatización de algún proceso.

Por lo que se me ofreció la posibilidad de trabajar con una celda industrial en el laboratorio de la Universidad Politécnica de Catalunya. Esto supone una gran oportunidad para comprender el funcionamiento de una línea de producción. También me ayudara a poner en practica todos los conocimientos adquiridos a lo largo del grado universitario.

Comprender los buses de comunicación, como interaccionan distintos PLC a la vez relacionados con distintos protocolos de comunicación. Trabajar con muchos elementos de entrada y salida, y aprender a gestionarlos. Aprender a buscar maneras para obtener un proceso eficiente.

Todo esto son los distintos motivos para seleccionar este proyecto, básicamente aprender de un área que me llama mucho la atención para poder dedicarme en un futuro.

2 Consideraciones previas

2.1 Normativa

Todas las aplicaciones industriales están regidas por una serie de normas estandarizadas, en este proyecto se tiene constancia de las normas de seguridad y las normas de programación. A continuación, se mostrará de manera breve la normativa vigente en el área de la seguridad y programación de los PLCs.

- Seguridad

IEC 61508 es una normativa internacional estándar adoptada por la comisión internacional de electrónica (IEC), donde se contempla todos los aspectos de seguridad que deben cumplir los sistemas electrónicos programables aplicado a cualquier sector industrial¹.

- Autómatas programables

IEC 61131 es el conjunto de normas e informes técnicos adoptados por la comisión internacional de electrónica (IEC) con el objetivo de estandarizar los autómatas programables. La parte 3 (**IEC 61131-3**) es la encargada de estandarizar los lenguajes de programación de los PLC¹.

2.2 Controladores lógicos programables (PLC)

2.2.1 Definición

Un controlador lógico programable más conocido por sus siglas en inglés PLC (programmable logic controller), es una computadora industrial capaz de automatizar todo tipo de proceso industrial. Dicha computado gestiona las entradas para procesar las salidas con el proceso automático deseado.

Para que un PLC controle y procese una tarea industrial previamente se debe programa para que realice la tarea deseada. Dependiendo de la marca del PLC dispondrá de un software. Cada programa dispondrá de diversos leguajes de programación donde programaremos nuestro proceso.

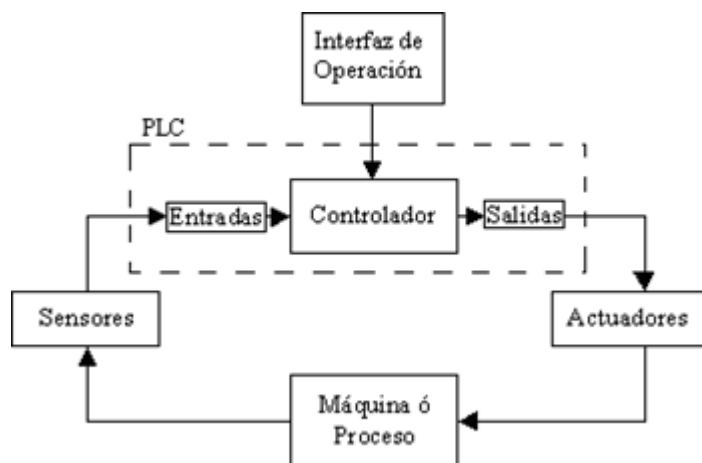


Figura 3 - Flujo funcionamiento del PLC

2.2.2 Estructura

Los PLC están compuestos por distintas partes, las cuales pueden estar integradas o por módulos.



Figura 4 - Ejemplo PLC

Fuente de alimentación

Su función es suministrar energía a la CPU y a las demás tarjetas del PLC.

CPU

Es la unidad central de procesamiento y se encarga básicamente de procesar las instrucciones que tiene programado el PLC.

Módulos de entrada y salida

Estos módulos son la conexión directa entre el PLC y el sistema de control. Los módulos de entrada se encargan de transmitir la señal de procedente de nuestro sistema al PLC. Estas señales de entrada proceden de sensores, pulsadores, finales de carrera, llaves etc. El módulo de salida se encargar de transmitir a los actuadores la señal proveniente del PLC. Las señales de entradas y salidas pueden ser digitales y analógicas.

Módulos de memoria

Módulos donde se almacena el programa del PLC. Existen dos tipos de memoria:

- Memoria no volátil
 - Memoria donde se almacena los programas que coordinan los recursos del equipo y los datos de manera permanente. Es solo de lectura.
- Memoria volátil
 - Memoria donde se almacena y ejecuta el programa utilizado.

2.2.3 ¿Por qué se utilizan los Controladores lógicos programables?

Los PLC son utilizados en prácticamente todas las aplicaciones industriales debido a una serie de ventajas.

- Reduce el tiempo de elaboración de proyectos, gracias a su facilidad a la hora de instalarlo y a su flexibilidad.

- La posibilidad de añadir modificaciones sin cambiar el cableado ni añadir elementos.
- Capacidad de evitar interferencia gracias a la inmunidad al ruido eléctrico.
- Grand rapidez de respuesta. Esto hace posible seguir proceso a tiempo real.
- Reducción de costos, ya que con un mismo PLC podemos adaptarlo para trabajar en otro proceso una vez finalizado con este.
- Facilidad a la hora de programar un PLC ya que el lenguaje que emplea es comprensible.
- Grand adaptación a los entornos industriales, un amplio rango de temperatura, resistencia a las vibraciones y al impacto.

2.2.4 Ciclo de funcionamiento de un PLC

Al iniciarse un PLC verifica una serie de elementos antes de ejecutar el programa del usuario. Verifica el estado del hardware, es decir, el estado de la CPU, la fuente de alimentación u otros módulos. También verifica la situación de la batería de respaldo y la integridad del programa del usuario. Posteriormente pone a cero todas las posiciones del registro de imagen e inicializa el registro de datos que consiste en establecer unas condiciones iniciales para los controladores, temporizadores y demás registros del sistema. Todas estas acciones solo las realiza una vez encendido el PLC.

En cuanto a la ejecución del programa, el PLC entra en proceso cíclico. Cada uno de estos ciclos son denominados SCAN. Primero realiza una lectura de los módulos de entrada y actualiza el registro de imagen de entrada. Posteriormente ejecuta el programa instrucción a instrucción, verifica los contactos según el registro de entrada y salida, después según la lógica implementada determina el estado de cada salida. Para acabar escribe los nuevos valores en el registro de salida. Finalmente, trámite los valores del registro a los módulos de salida.

Finalizado un ciclo de SCAN se inicia otro de nuevo empezando por el primer paso. Estas operaciones se repetirán una y otra vez mientras se encuentre en operación.

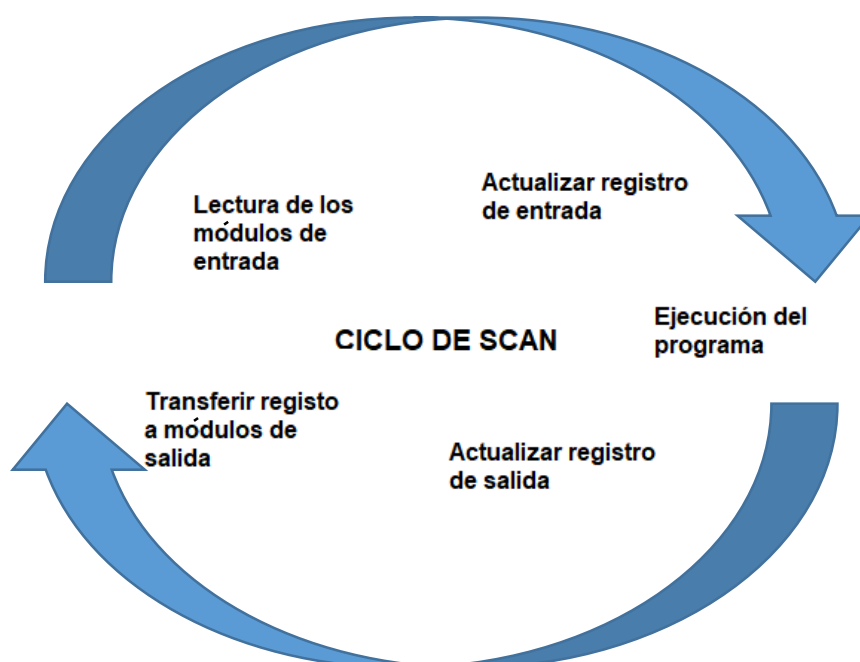


Figura 5 - Ciclo de SCAN

2.3 Protocolos de comunicación

2.3.1 Protocolo Profibus

Según la información obtenida en la aula21², Profibus es un protocolo propiedad de la organización Profibus International. Su nombre procede de Process Field BUS.

Profibus se basa en la norma RS485 y puede admitir un máximo de 32 nodos. No obstante, este límite puede ser ampliado hasta 127 nodos mediante el uso de repetidores, pero solo pueden ser nodos activos un máximo de 32 y cada repetidor cuenta como un nodo.

El intercambio de mensajes tiene lugar en ciclos donde una estación maestra envía una trama a una estación maestra o esclava que reconoce y envía la respuesta. Esto se produce cuando una estación maestra está en el intervalo de posesión del testigo o token, ya que considera a todas las otras como esclavas al estar en posesión del testigo y tener acceso al bus de comunicación.

2.3.2 Protocolo CAN

Partiendo de las explicaciones de National Instruments³, originalmente, CAN (Controller Area Network, por sus siglas en inglés) fue desarrollado por Bosch en 1985 para redes en vehículos. Es un bus de área de control que actualmente se utiliza en muchos campos (tranvías, subterráneos, trenes ligeros, trenes de distancias largas, industria, etc.).

Las principales características son:

- Capacidad de dar prioridad a los mensajes de la trama.
- Formado por 4 hilos, dos pares de cables trenzados.
- 64 nodos como máximo por segmento y 127 como máximo en todo el bus.
- Detección y señalización de errores.
- Control de errores avanzando, distinción entre errores temporales y permanentes.
- Recepción por multidifusión (multicast) con sincronización de tiempos.
 - **Productor-consumidor:** comunicación escrita-solicitud de eventos
 - **Maestro-esclavo:** el maestro hace una petición y el esclavo direccionando responde.

La impedancia del bus es de 60 Ω y lleva una resistencia terminadora de 120 Ω en cada extremo. El nivel de voltaje en los cables varía entre 1,5V a 3,5V (CAN_H: 2.5 a 3.5V; CAN_L: 2.5 a 1.5V).

2.3.3 Ethernet

Partiendo de las explicaciones del Prof. Ing. Diego M: Romero⁴, Ethernet es un estándar de redes de área local con acceso al medio CSMA/CD. Ethernet define las características del cableado y señalización del nivel físico y las fórmulas de trama de datos del nivel de enlace del modelo OSI. Utiliza dos tipos de datagramas, el Ethernet II y el Standart IEE802.3, la diferencia entre estos dos tipos de datagramas está en dos bytes.

Se utiliza cable coaxial, par trenzado o fibra óptica para la transmisión de los datos dependiendo del uso de la red. Cada tipo de cable permite una velocidad de transmisión y una longitud máxima de cable.

Para identificar tanto la dirección como la aplicación a la cual se dirige un paquete determinado se utilizan dos números la IP y el Puerto.

2.4 Software de programación

Unity Pro XL es el software encargado de la programación de los autómatas de la marca Schneider Electric. Es un programa capaz de ejecutar las instrucciones del proyecto generado. El software es compatible con los modelos Modicom M340, Premium, Quantum y Atrium.

Unity Pro XL se basa en la norma, IEC 61131-3. Esta norma define los lenguajes de programación de uso más corriente, las reglas sintácticas y semánticas y el juego de instrucciones fundamentales, etc.

Gran parte de la información de este apartado ha sido extraída del Manual de UNITY⁵.

2.4.1 Lenguajes de programación

Unity Pro XL tiene a la disposición del usuario cinco tipos de lengua de programación, esto nos permite una gran flexibilidad a la hora de programa. Dependiendo del caso es mejor elaborar el código con un lenguaje u otro, también depende del usuario y del lenguaje que le parezca más óptima para su uso. Todos estos lenguajes están estandarizados según la norma IEC 61131-3.

Diagrama de Contactos (Ladder)

Se llama diagrama de escalera ya que presenta una estructura muy similar, dos riales verticales de alimentación y varios horizontales donde se encuentra la lógica del circuito. Al lado izquierdo se colocan las instrucciones y al lado derecho las salidas. El PLC hace una lectura de arriba a abajo y de izquierda a derecha.

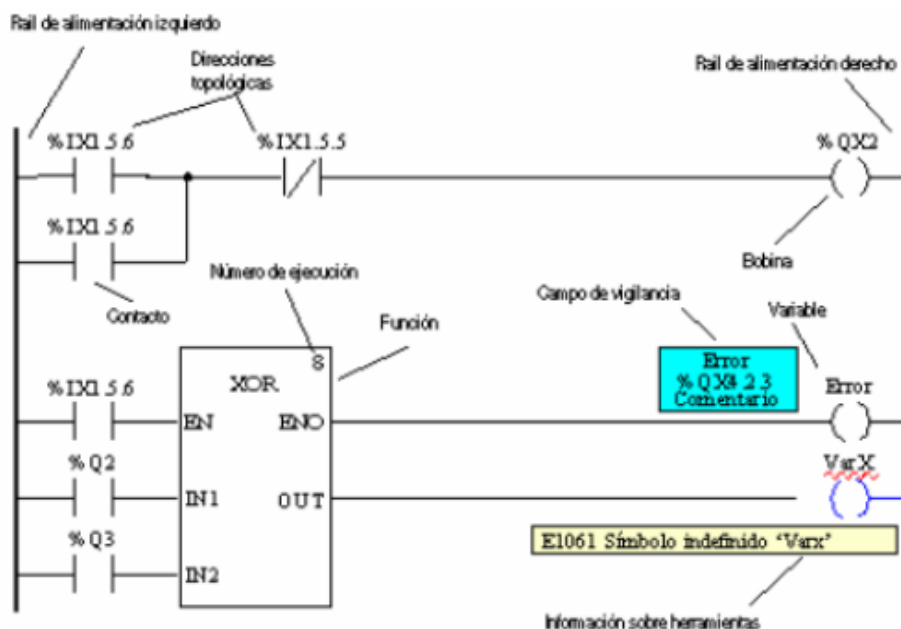


Figura 6 - Diagrama de Contactos (Ladder)

Diagrama de bloques de función – FBD

En este tipo de lenguaje se utilizan bloques de símbolo lógico, es decir basado en puertas lógicas.

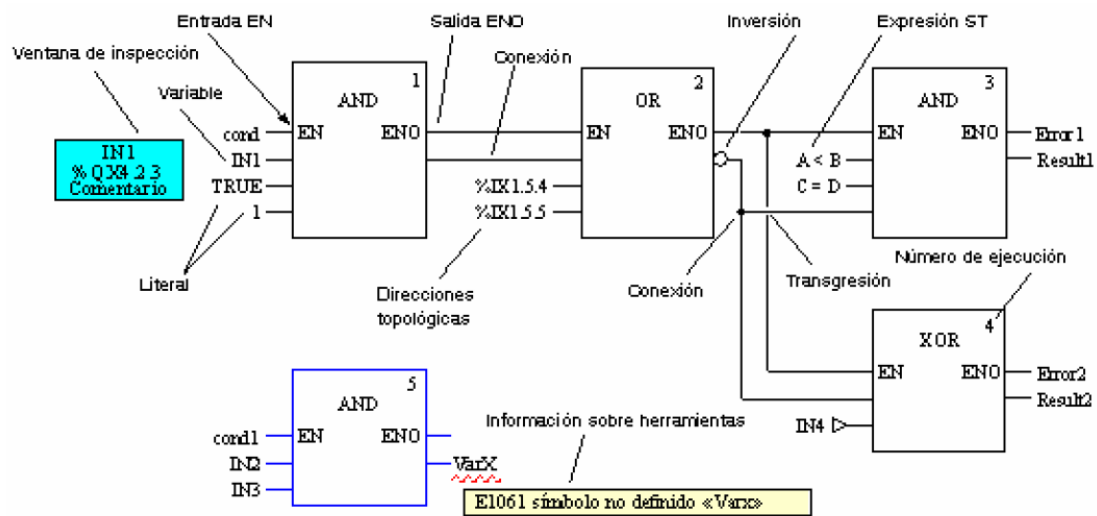


Figura 7 - Ejemplo FBD

Lista de Instrucciones-IL

Es el lenguaje más antiguo de ellos, se trata de un lenguaje de texto basado en ensamblador, es decir de bajo nivel. Es un tipo de lenguaje compacto y completo ya que ejerce un control directo sobre el hardware.

Una de las características principales es que trabaja con un registro especial, denominado acumular y es conveniente para programas de poca extensión.

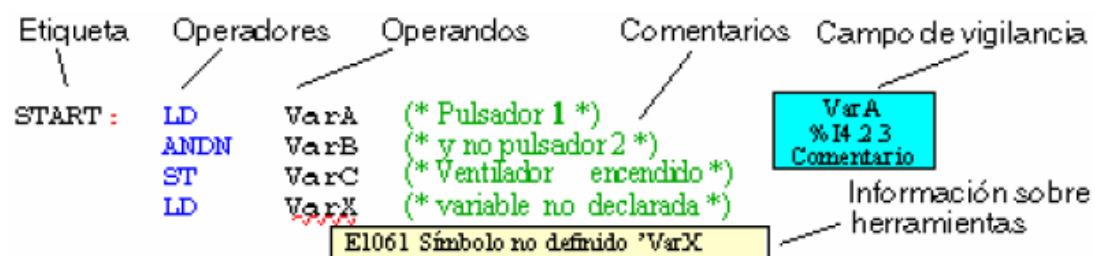


Figura 8 - Ejemplo Lista de Instrucciones

Texto estructurado – ST

Se trata de un lenguaje de alto nivel muy similar al Pascal y C, este sistema de programación se lleva a cabo mediante series de instrucciones concretas que pueden utilizarse dentro de su entorno. Dispone de estructuras para bucles, ejecución condicional y una serie de funciones.

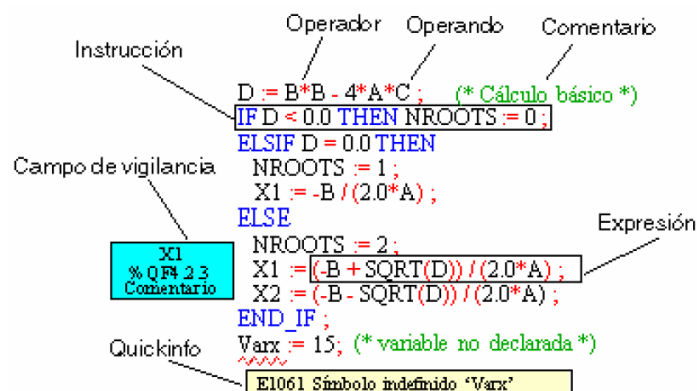


Figura 9 - Texto estructurado – ST

Diagrama Secuencial – Grafcet – SFC

El GRAFCET es un diagrama funcional que describe los procesos a automatizar, teniendo en cuenta las acciones a realizar, y los procesos intermedios que provocan estas acciones.

Un GRAFCET está compuesto de:

- **ETAPA:** define un estado en el que se encuentra el automatismo. Las etapas de inicio se marcan con un doble cuadrado.
- **ACCIÓN ASOCIADA:** define la acción que va a realizar la etapa, por ejemplo, conectar un contacto, desconectar una bobina, etc.
- **TRANSICIÓN:** es la condición o condiciones que, conjuntamente con la etapa anterior, hacen evolucionar el GRAFCET de una etapa a la siguiente, por ejemplo, un pulsador, un detector, un temporizador, etc.

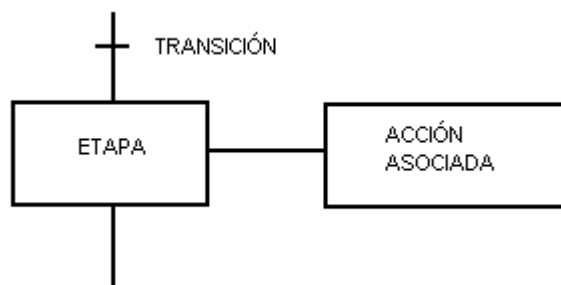


Figura 10 - Ejemplo de Etapas Grafcet

2.4.2 Sección de programación

Las secciones de programación son entidades autónomas que permiten programar en los distintos lenguajes que existe en el Unity Pro (LD, FBD, IL, ST, SFC). Por lo tanto, dentro de un mismo programa podemos tener distintas secciones con diferentes lenguajes de programación.

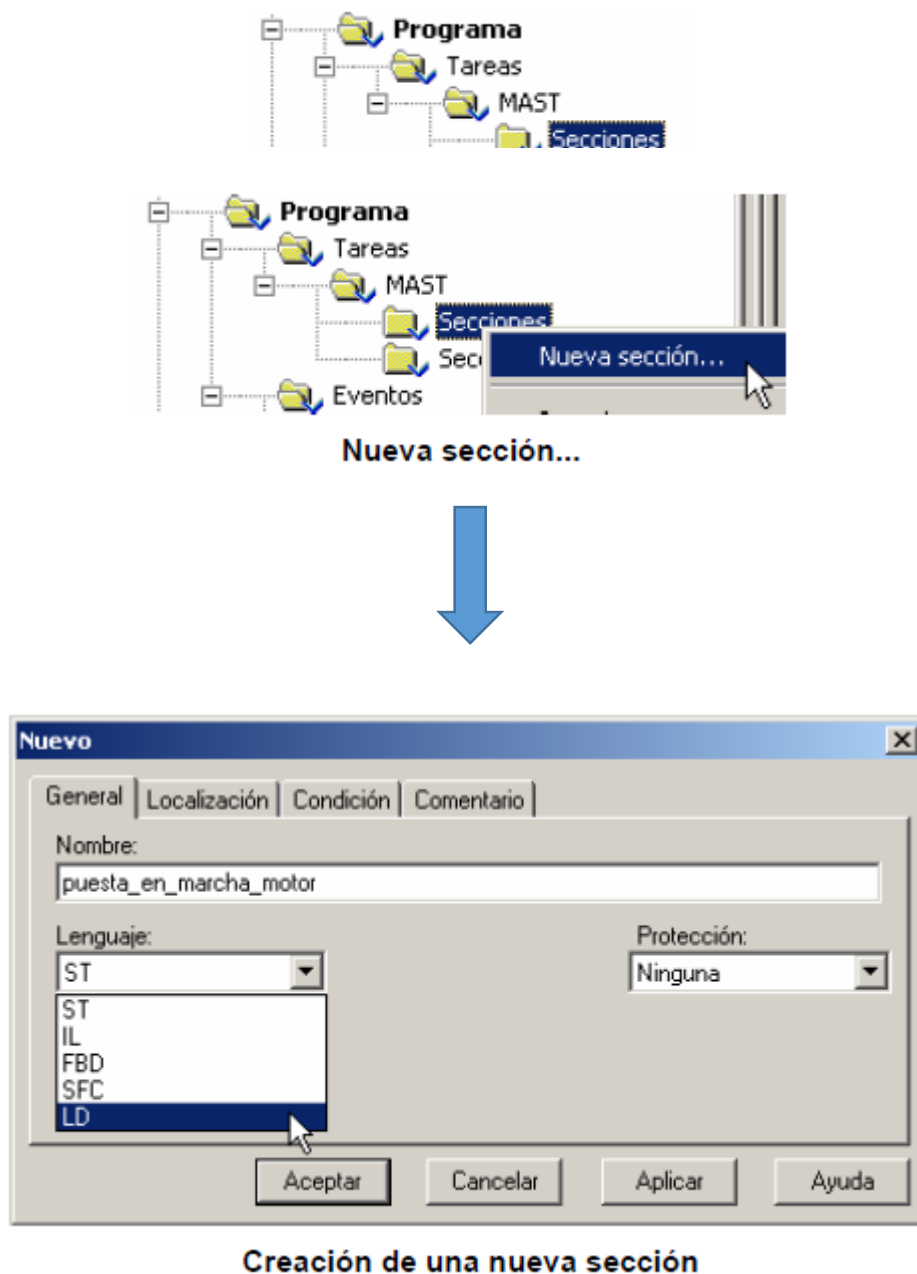


Figura 12 - Crear una nueva Sección

Para crear una sección nueva, desde el Explorador de Proyectos, hacemos un clic en el botón derecho y seleccionamos Nueva sección... finalmente se elige el nombre de la sección y el tipo de lenguaje con el cual deseas programar.

Como hemos dicho anteriormente es posible crear varias secciones, por lo que es importante tener en cuenta orden de ejecución. Este orden vendrá definido por la disposición de las secciones, es decir, en función del orden de las secciones el código de programación será eficaz o erróneo. El orden de las secciones se puede modificar arrastrando a la posición deseada.

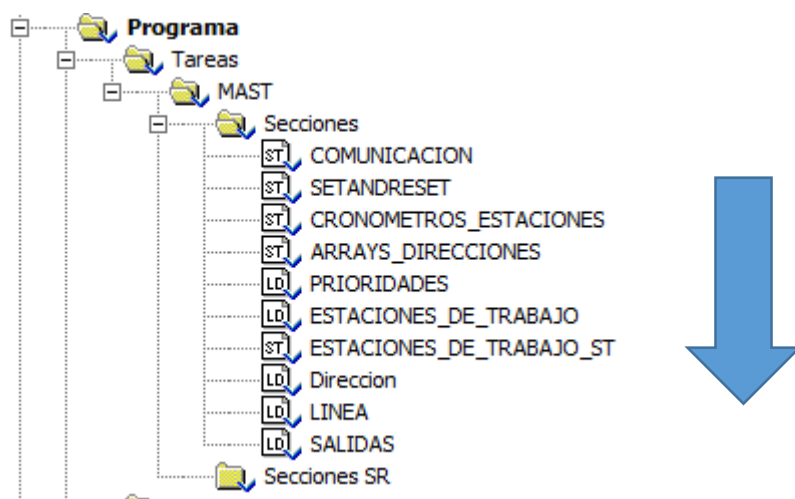


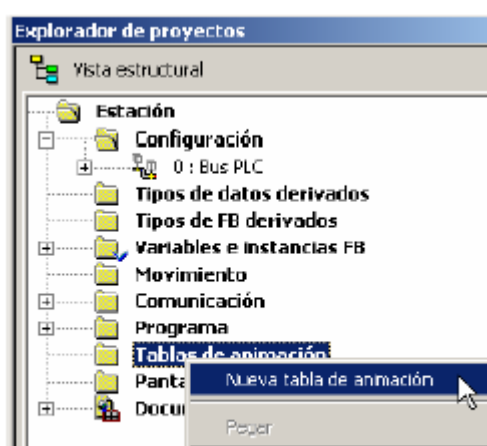
Figura 13 - Orden de las Secciones

2.4.3 Tabla de animación

Las tablas de animación son ventanas que nos permiten visualizar el valor de las variables (0 o 1 si son señales digitales, valor numérico si son enteros o reales), también permite modificar o forzar los valores de dichas variables.

La tabla de animación es una gran herramienta para comprobar el correcto funcionamiento de tu proceso.

Para crear una tabla de animación vamos al Explorador de proyectos, a la carpeta de Tablas de animación, hacemos clic derecho y seleccionamos Nueva tabla de animación.



Creación de una tabla de animación

Figura 14 - Crear Tabla de animación

Tabla de animación a tiempo real:

Nombre	Valor	Tipo
escrituraRFID02	987	INT
escrituraRFID03	878	INT
%MW2	8	INT
%MW3	8	INT
M100	0	EBOOL
M101	0	EBOOL
M102	0	EBOOL
M103	1	EBOOL
M104	0	EBOOL
M105	0	EBOOL
M106	0	EBOOL
M107	1	EBOOL

Figura 15 - Ejemplo real Tabla de animación

En negrita observamos los valores que podemos forzar (1 o 0) o modificar:

Modificación	Forzar				
Nombre	Valor	Tipo	Nombre	Valor	Tipo
DIR17_REST	0	EBOOL	DIR17_REST	0	EBOOL
DIR17_READY	1	EBOOL	DIR17_READY	1	EBOOL
DIR17_MOVE	0	EBOOL	DIR17_MOVE	0	EBOOL
DIR15_REST	0	EBOOL	DIR15_REST	0	EBOOL
DIR15_READY	1	EBOOL	DIR15_READY	1	EBOOL
DIR15_MOVE	0	EBOOL	DIR15_MOVE	0	EBOOL
PT12_REST	1	EBOOL	PT12_REST	1	EBOOL
PT12_READY	0	EBOOL	PT12_READY	0	EBOOL
PT12_MOVE	0	EBOOL	PT12_MOVE	0	EBOOL
DIR18_REST	0	EBOOL	DIR18_REST	0	EBOOL
DIR18_READY	0	EBOOL	DIR18_READY	0	EBOOL
DIR18_MOVE	1	EBOOL	DIR18_MOVE	1	EBOOL
PT13_REST	1	EBOOL	PT13_REST	1	EBOOL

Figura 16 - Ejemplo real Tabla de animación

Modificar:



Forzar:



2.4.4 Pantallas de operador

Las pantallas del operador son ventanas que nos permiten insertar botones, textos, indicadores, números, imágenes desde la librería. Esta herramienta ayuda tener tu proceso de manera visual.

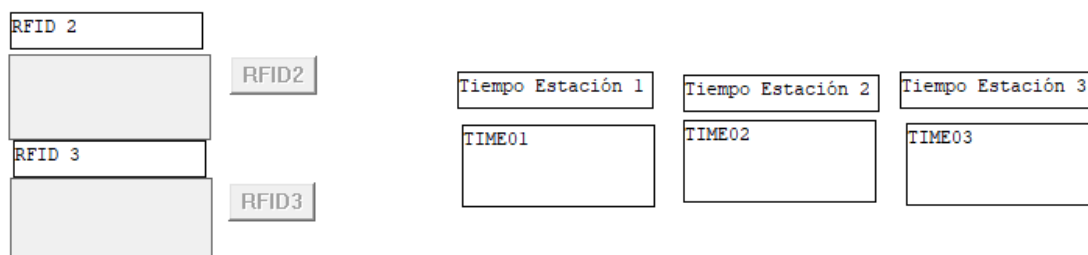


Figura 17 - Ejemplo Pantallas de operador

Para crear una pantalla de operador son dirigimos al Explorador de proyectos, hacemos clic en el botón derecho sobre la carpeta de Pantallas de operador y seleccionamos Nueva pantalla.

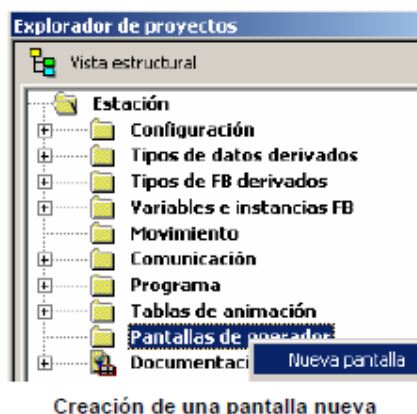


Figura 18 - Crear Nueva Pantalla operador

2.4.5 Tipos de datos y variables

El Unity Pro nos permite la posibilidad de crear distintos datos. A la hora de programa hemos utilizado una serie de variables que explicaremos a continuación. Primero del todo hay que diferencia entre una variable alcatada y una variable que no lo es.

Una variable alcatada también conocida como direccionada está asociada a una dirección de memoria, es decir, conocemos su posición de memoria. En cambia una variable no alcatada no está asociada a una referencia de memoria, por lo tanto, no es posible conocer la posición que ocupa en la memoria.

Nombre	Tipo	Dirección
REARME	EBOOL	%I0.1.6
RESET	EBOOL	%M21
seguridad	EBOOL	

Figura 19 - Ejemplo variable Alcatada y no Alcatada

En la figura 18, podemos observar que la variable REARME y RESETE tienen una dirección asociada, en cambio la variable seguridad no.

En el Unity existen tres tipos de datos.

EDT (Elementary Data Type)

- **BOOL:** De tipo booleano, es decir, TRUE (1) o FALSE (0). Ocupa un byte de memoria, pero el valor solo se guarda en un bit y el valor predeterminado es el FALSE.
- **EBOOL:** De tipo booleano, pero a diferencia de las variables BOOL, esta es capaz de gestionar los flancos ascendentes o descendientes. También ocupa un byte de memoria. El valor lo guarda en un bit, en otro bit guarda el registro para gestionar los flancos y otro bit donde contiene el estado de forzado.

- INT: variables de tipo ENTERO con signo y forma y ocupa 2 bytes (-32768 hasta 32767).
- TIME: tipo de variable que se representa con un ENTERO doble sin signos, (UDINT). Esta variable representa el tiempo en milisegundos con una duración máxima de aproximadamente 49 días. Las unidades del tiempo permitidas son días (D), horas (H), minutos (M), segundos (S) y milisegundos (MS).

Para crear una variable elemental nos dirigimos al Explorador de proyectos. Seleccionamos la parte de Variables elementales, una vez dentro observaremos las variables creadas previamente. Finalmente hacemos clic doble en la última fila y le asignamos un nombre.

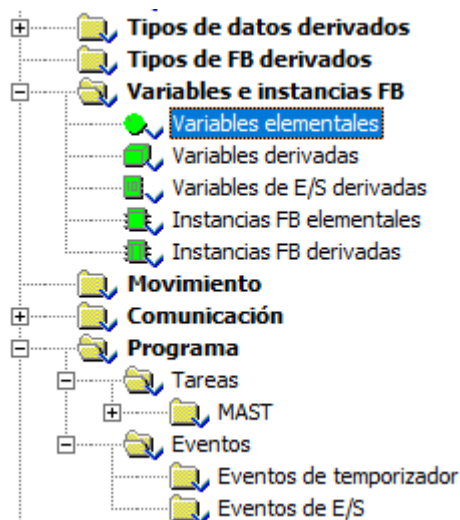


Figura 20 - Variables elementales

DDT (Derived Data Type)

Estas variables son un tipo de dato derivado (estructura o arrays). Para crear un tipo de dato derivado de tipo estructurado, se debe acceder a Variables e instancias FB, seleccionar el comando Abrir del editor de datos y seleccionar la ficha Tipos de DDT.

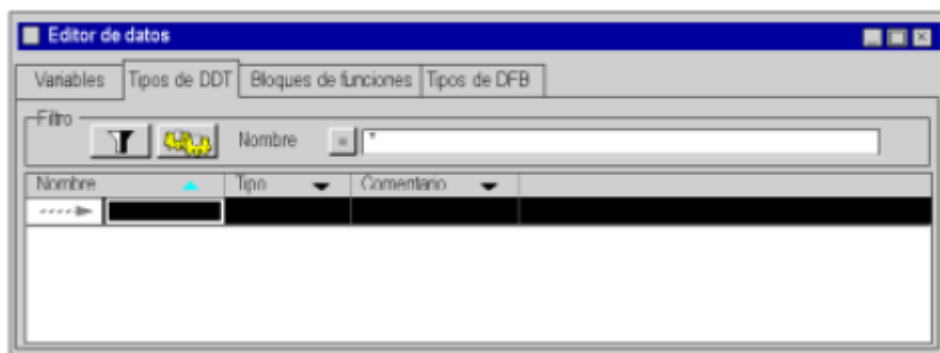


Figura 21 - Crear Derived Data Type

IODDT (Input Output Derived Data Type)

Son las variables de diagnóstico relacionadas con módulos de hardware. Este tipo de variable no se ha utilizado en la elaboración de este trabajo.

2.4.6 Biblioteca de funciones

En los lenguajes de programación es posible insertar bloques de función. En los lenguajes en de tipo grafico (LD y FBD) aparece en formato de bloques y en los leguajes de tipo texto (ST y IL) aparece en formato texto/estructurado. Los bloques de función vienen de un conjunto de librerías y dentro de cada librería se encuentran familias.

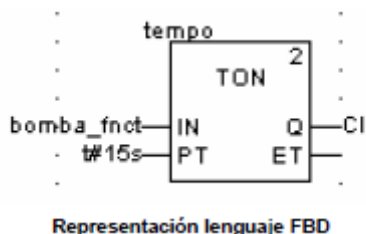


Figura 23 - Representación lenguaje FBD

```
(*temporizador de 15s*)
TEMPO (IN := bomba_fnct,
      PT := time#15s,
      Q => CI);
```

Representación lenguaje ST

Figura 22 - Representación lenguaje ST

Función elemental

Las funciones elementales (EF) no disponen de estado interno y sólo cuentan con una salida. Si en las entradas aparecen los mismos valores, siempre que se ejecute la función el valor de la salida será el mismo, por ejemplo, siempre que se ejecuta la suma de dos valores el resultado es el mismo. El número de entradas puede aumentarse con algunas funciones elementales.⁵

Bloque de funciones elementales

Los bloques de funciones elementales (EF) tienen estados internos. Si las entradas disponen del mismo valor, el valor de la salida puede variar cada vez que se ejecuten los bloques de funciones. Por ejemplo, con un contador aumenta el valor de la salida. Se utilizan instancias.⁵

Bloques de funciones derivados

Los bloques de funciones derivados (DFB) presentan las mismas propiedades que los bloques de funciones elementales. Sin embargo, el usuario los crea en los lenguajes de programación FBD, LD, IL o ST.⁵

2.4.7 Tipos de direcciones

tipos de direcciones e información que gestiona y transmite. Las direcciones siempre van expresadas con un (%) seguido de uno o dos letras en función del tipo de dirección. Dichas direcciones están regidas por la norma IEC 61121-3.

%I	Entrada digital
%Q	Salida digital
%IW	Entrada analógica
%QW	Salida analógica
%M	Bit de memoria (0 o 1)
%MW	Palabra de memoria (16 bits)
%KW	Palabra constante (no modificable durante la ejecución)
%S	Bit de sistema
%SW	Palabra de sistema

Tabla 1 – Direcciones

2.4.8 Bit de sistema

El bit de sistema **%Si**, te permite visualizar el estado de los autómatas y controlar su funcionamiento.

En nuestro proyecto utilizamos el bit de sistema %S13 (1RSTSCANRUN).

Este bit se pone a 1 en el primer ciclo después de ponerse en RUN, es decir, en el proceso que el PLC pasa de STOP a RUN, este bit se pone a 1 durante un ciclo de SCAN y vuelve a su estado inicial (0).

3 Celda de trabajo

3.1 Introducción

La celda industria del laboratorio nos permite transportar bandejas, en nuestro caso simularan los paquetes. Estas bandejas se transportan mediante correas de distribución accionada por motores.

Los retenedores se encargan de detener las banderas y las plataformas de cambiar su dirección. También disponemos de un pulmón que se encarga de almacenar las bandejas, pinzas para extraer productos y pinzas para coger las bandejas y transportarlos. Todos estos elementos irán gestionados por los distintos PLCs de la celda.



Figura 24 - Celda Industrial

La celda industria está dotada de 4 tipos de comunicaciones: Profibus, CAN, Ethernet y ASI-INTERFACE.

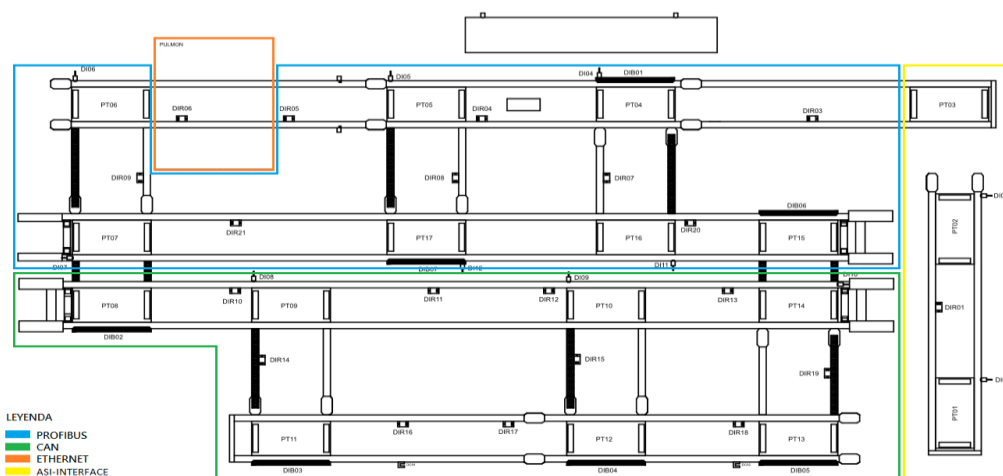


Figura 25 - Distribución de los protocolos

En nuestro proyecto hemos utilizado elementos de las líneas Profibus, CAN y Ethernet. En las líneas CAN y Profibus se han utilizado los retenedores y las plataformas, y en la línea Ethernet el sensor (DIR06).

La distribución de la celda industria del laboratorio permite gestionar de manera independiente cada zona de la celda, lo que proporciona una gran ventaja, ya que tienes una visión más detallada de cada zona. También te permite situar los periféricos cerca del sistema de gestión y conectarlo con un mínimo de cables al dispositivo maestro (PLC) utilizando el protocolo de comunicación adecuado en cada caso.

Esto también tiene sus inconvenientes ya que cada zona tiene que recibir la información de la otra zona de trabaja, y a ser capaz de interconectarlas de forma correcta.

Todos los PLC de la celda se comunican entre ellos mediante la capa física Ethernet.

3.2 Componentes de la celda

La celda industrial del laboratorio está compuesta por una serie de elementos que hacen posible que las bandejas se desplacen de forma correcta.

Sensores inductivos

los sensores de proximidad inductivos detectan objetos metálicos que se aproximan al sensor, sin tener contacto físico con los mismos. el sensor crea un campo magnético de alta frecuencia que es generado por la bobina, cuando el objeto metálico se acerca la oscilación se atenúa o decrece. Este cambio es detectado por el sensor y es cuando emite una señal de detección.

En la celda industrial hay tres tipos de detectores inductivos.

Los inductivos que al llegar la bandeja a la línea detectan una pieza metálica que hay en el lateral cuando la plataforma neumática está bajada.



Figura 26 - Sensor inductivo

Los inductivos basculantes que se encargan de detectar la llegada o salida de las bandejas a la plataforma neumática. Su funcionamiento se basa en el movimiento de un basculante que activa o desactiva un sensor inductivo.

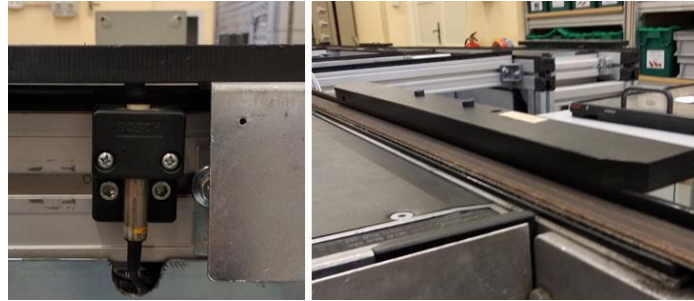


Figura 28 - Sensor inductivo basculante, no detecta llegada bandeja



Figura 28 - Sensor inductivo basculante, detecta llegada bandeja

Los inductivos que detectan la llegada de la bandeja gracias a una pieza metálica que está situada sobre un soporte de plástico.



Figura 29 - Sensor inductivo retenedor y pieza metálica detectada

Sensores capacitivos

Los sensores capacitivos son capaces de detectar objetos metálicos y no metálicos sin un contacto físico. Al aproximarse a la superficie activa sobrepasan una determinada capacidad.

Este sensor no se ha utilizado en la simulación del nuestro proceso por el gran inconveniente que supondría a la hora de programar.

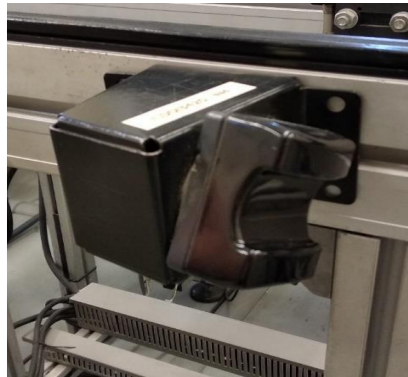


Figura 30 - Sensor capacitivo

Sensores fotoeléctricos de fibra óptica

Los sensores fotoeléctricos detectan la presencia de objetos situados sobre las bandejas. El sensor emite una luz que viaja a través de la fibra y regresa al amplificador. El amplificador decide si la cantidad de luz sobrepasa el umbral programado para activar una salida.

Este sensor so ha sido utilizado en la simulación del proceso.



Figura 31 - Sensores fotoeléctricos de fibra óptica

Electroválvula o válvula solenoide

Una electroválvula es un elemento cuya función es regular el flujo de un fluido. Este tipo de válvulas se mueve por la acción de una bobina solenoide.

Las electroválvulas son las encargadas de accionar los retenedores y las plataformas permitiendo el paso o no de aire comprimido de acuerdo con el programa definido por el usuario.



Figura 32 - Electroválvula o válvula solenoide

Plataformas

Las plataformas son los elementos situados en las intersecciones de la línea de producción y permiten la conexión con otras líneas de producción. Por lo tanto, las plataformas interconexiones las líneas de producción y permiten el flujo correcto de las bandejas.

En la celda industria hay dos tipos de plataformas según sus posibles estados.

- Primer tipo es capaz de adoptar dos estados: posición de reposo y subida.
- Segundo tipo capaz de adoptar tres estados: posición de reposo, subida y bajada.



Figura 33 - Plataforma en estado de bajada, de reposo y subida

Cada estado realiza una función, el estado de reposo no permite que la bandeja circule. El estado de bajada libera la circulación, es decir, permite que la bandeja circule. Y el estado de subida hace posible el movimiento entre líneas con diferentes alturas.

Retenedores

La función del retenedor es permitir la circulación de la bandeja. Es el encargado de retenerla o dejarla circular. Está compuesto por un sensor inductivo que detecta la presencia de la bandeja y por un cilindro simple. Son accionados por electroválvulas.



Figura 34 - Retenedor

Pulsadores sin enclavamiento

Un pulsador es un tipo de interruptor que solo realiza su función mientras está presionado, es decir, sin enclavamiento. La función del pulsador es permitir o cortar el paso de la corriente. Existen dos tipos de pulsadores dependiendo del funcionamiento. NC (NC) y NA (NO), es decir normalmente cerrados y normalmente abiertos.

En nuestro proyecto no se han utilizado.

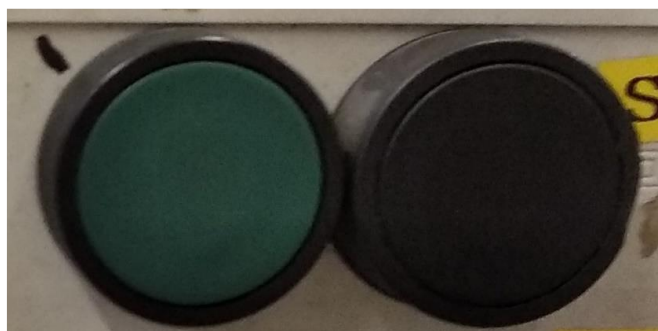


Figura 35 - Pulsador

Seta de emergencia

Según la norma EN ISO 13850, la función de parada de emergencia sirve para prevenir situaciones que puedan poner en peligro a las personas, para evitar daños en la máquina o en trabajos en curso o para minimizar los riesgos ya existentes, y ha de activarse con una sola maniobra de una persona.⁶



Figura 36 - Seta de emergencia

Motores cintas transportadoras

Son los motores encargados del movimiento de las cintas transportadoras de la línea industrial. Se tratan de motores trifásicos conectados en estrella

Motores de la marca BOSCH.



Figura 37 - Motor BOSCH

Contactor

Un contactor es un elemento electromecánico capaz de permitir o cortar la corriente eléctrica de una carga. Es una especie de interruptor, pero un contactor es capaz de realizar la acción a distancia mediante un elemento de comando.

4 Tecnologías de identificación

4.1 Introducción

Para el reconocimiento o identificación de nuestro producto es necesario algún elemento que nos permita realizar dicho reconocimiento. A continuación, se presentará una serie de tecnologías usadas en el mercado, con la finalidad de decidir cuál de ellas se adapta mejor a nuestras necesidades.

Tecnología RFID

La tecnología RFID es un sistema de almacenamiento y recuperación de datos, basada en un chip (etiquetas, tarjetas o transpondedores RFID), permite identificar de manera remota el objeto en el que se coloca mediante una señal de radio.

La principal ventaja de un RFID es la identificación. Se realiza mediante ondas de radio, esto permite que se pueda acceder a la información del chip sin la necesidad de que haya contacto visual o física. Además, en los últimos años ha adquirido relevancia por el abaratamiento de su coste.

Código de barras

El código de barras es un código basado en la representación de líneas paralela con diferente grosor y espacio. Este conjunto de barras y espacios representan una cadena de caracteres. De esta manera el código de barras permite identificar rápidamente el producto de manera única y global.

Para la lectura de un código de barras tiene que haber contacto directo.

Las principales ventajas son la rapidez de lectura, el bajo coste y la alta fiabilidad.

Código QR

El código QR es modulo para almacenar información en una matriz de puntos o en un código de barras bidimensional. Dicho código es la evolución de código de barras. La única diferencia respecto al código de barras es la cantidad de información que pueden condesar. En este caso el código de QR es capaz de almacenar más información.

La principal ventaja respecto a un código de barras es la capacidad de almacenamiento.

Cámaras de identificación

Este sistema está basado en una serie de cámaras capaces de reconocer una serie de objeto, gracias al sistema de evaluación que incorporan.

4.2 Elección de la tecnología de reconocimiento (caso de estudio)

Criterio	RFID	Código barras	Código QR	Cámaras de reconocimiento
Coste	Medio	Bajo	Bajo	Alto
Ubicación de ítems	No provee la ubicación física absoluta	Identifica el ítem en un punto específico en la línea	Identifica el ítem en un punto específico en la línea	Identifica el ítem en un punto específico en la línea
Instalación	Sencillo	Sencillo	Sencillo	Muy complejo
Seguridad	Alto(código de acceso)	Baja (información en la parte inferior)	Baja-media	Alta
Almacenamiento Datos	Varios ítems	Solo un ítem	Varios ítems	----- --
Velocidad lectura	Mayor rapidez de lectura (pallet completo)	Menor rapidez de lectura (caja por caja)	Menor rapidez de lectura (caja por caja)	Menor rapidez de lectura (caja por caja)
Modo de lectura	No requiere línea de visión	Requiere verificación visual directa	Requiere verificación visual directa	Requiere línea de visión
Flexibilidad de información	Actualización en tiempo real (lectura/escritura) simultanea	Poca flexibilidad (generación de nuevos códigos)	Poca flexibilidad (generación de nuevos códigos)	Actualización en tiempo real
Precesión	Alta	Baja (código perfectamente orientado, para su lectura)	Baja (código perfectamente orientado, para su lectura)	Alta
Durabilidad	Limitado por la unión entre la antena y chip	Cierto grado de degradación	Cierto grado de degradación	Alta

Tabla 2 - Tecnología de identificación (Criterios)

Analizado los diferentes criterios podemos llegar a una conclusión sobre la tecnología más óptima para nuestra línea de producción. Después de observar esta tabla de características podemos descartar el código de barras y el código QR, ya que consideramos muy importante el aspecto de la lectura. Como ya hemos comentado antes, esta tecnología necesita visión directa con el código para leer la información del paquete. Esto nos lleva a pensar en los inconvenientes que supondría esta restricción. Si queremos que el paquete llegue a ser identificado, el producto deberá llegar totalmente alineado, para que el lector sea capaz de captar la información. Esto es un gran problema, porque no podemos asegurar que la caja llegue alineada con el lector.

Sí que es verdad que estas tecnologías son más económicas, además son muy sencillas y fáciles de instalar, pero como hemos comentado anteriormente consideramos de gran importancia la correcta identificación de nuestros productos.

Después de descartar estas dos tecnologías solo nos queda la RFID y las cámaras de reconocimiento. Analizando dichas tecnologías observamos que las dos son capaces de leer la información independientemente de la orientación del paquete. Comparando los demás criterios vemos que en la mayoría de ítems las dos técnicas son muy similares y además cumplen con éxito las necesidades de nuestra línea de producción. Esto nos ha llevado a analizar los puntos que las diferencian; coste, durabilidad y complejidad.

La tecnología RFID es más económica que las cámaras de reconocimiento en todos los aspectos, ya sea a la hora de instalarlas o a la hora de llevar cabo el mantenimiento. Las cámaras incorporan un software que tendremos que adaptar a nuestro proceso y esto hace que tenga una mayor complejidad respecto a las RFID.

También hay que comentar que las cámaras tendrán mucha dificultad para diferenciar las cajas, que normalmente suelen tener la misma forma, aunque contengan diferentes productos en su interior. Sí que es verdad que se podría incorporar cualquier clase de distintivos a los paquetes para que las cámaras sean capaces de identificarlos, pero esto conlleva trabajo añadido que hace que esta tecnología pierda una de sus mejores cualidades, que suponía no perder tiempo a la hora de clasificar nuestras cajas.

La unión entre el chip y la antena es un problema que hay que tener en cuenta, ya que cuando la unión se rompe por cualquier problema tenemos un producto sin identificar en nuestra línea. Dicho problema no lo encontraríamos al usar las cámaras ya que no hay que incorporar nada que se pueda dañar al paquete.

En conclusión, después de observar estas diferencias nos decantamos por la tecnología RFID, ya que consideramos la opción más óptima para nuestra planta. También hay que decir que es una tecnología más extendida que las cámaras en el mundo de la paquetería. Esto quiere decir que a la hora de adquirir la tecnología disponemos de una mayor oferta. Creemos que es sencilla, pero a la vez la más eficiente para lo que buscamos.

4.3 Dispositivo RFID

4.3.1 Definición

RFID es un sistema de almacenamiento y recuperación de datos, basada en un chip (etiquetas, tarjetas o transpondedores RFID), permite identificar de manera remota el objeto en el que se coloca mediante una señal de radio.



Figura 38 - Componentes RFID

4.3.2 Estructura RFID

Un sistema RFID está básicamente compuesto por el lector y una etiqueta.

Lector

Los lectores son los encargados de enviar una señal de radiofrecuencia para detectar las posibles etiquetas dentro del rango de trabajo.

Existen dos tipos de lector según su fabricación, bobina simple y sistemas interrogadores con dos bobinas.

- Sistema con bobina simple, se usa la misma bobina para transmitir energía y datos. Son más baratas y simples, pero menor alcance.
- Sistemas interrogadores con dos bobinas, una bobina para transmitir energía la otra para los datos. Estas son más caras, pero obtenemos un mayor rango de alcance.



Figura 40 - Ejemplo Lector RFID



Figura 39 - Ejemplo lector RFID corte

Etiquetas RFID

Un tag, transpondedor o etiqueta electrónica son dispositivos pequeños, que pueden ser pegados o incorporados en el producto. Incorporan una antena y un microchip para ser capaces de gestionar las señales de radiofrecuencias desde un emisor receptor.

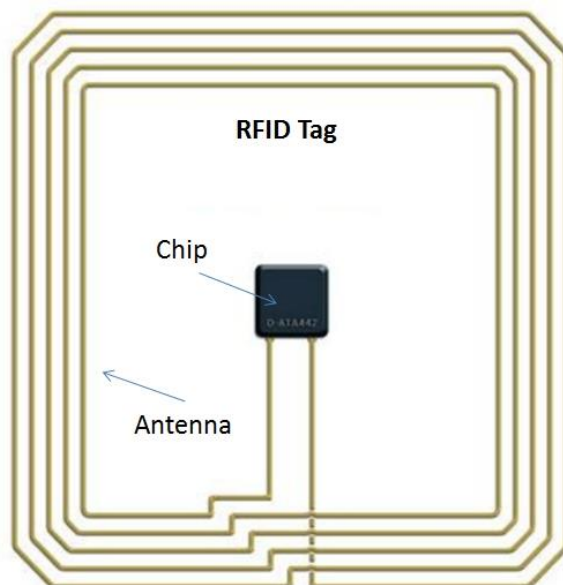


Figura 41 - Estructura de Etiqueta RFID

Alimentación

Las etiquetas o Tag necesitan muy poca energía para su funcionamiento, del orden de los mW. Dependiendo de la energía que utilizan para la comunicación podemos diferenciar 2 tipos de transpondedor:

- Etiquetas activas: Este tipo de etiquetas necesitan una batería adicional, ya que no tiene suficiente con la energía proporcionada por el lector. Posen la ventaja de tener un mayor alcance y además no es necesario que el lector sea quien inicie la comunicación.
- Etiquetas pasivas: no necesitan baterías adicionales para su funcionamiento, ya que únicamente se alimentan de la energía generada por el lector. También aprovechan la energía generada por el lector para transmitir al lector la información del microchip.

4.3.3 Opciones de programa

Los transpondedores se pueden dividir en dos tipos en función del tipo de memoria de la etiqueta, pueden ser solo de lectura, o de lectura/escritura. Las que solo son de lectura, suelen venir ya con su número de identificación por defecto. Pero normalmente ambos tipos pueden ser programados por el usuario.

4.3.4 Funcionamiento RFID

El funcionamiento del sistema está basado en captar la etiqueta y obtener la información que contiene. Como podemos observar en la figura 41, el lector envía una serie de ondas de radiofrecuencia al tag que son captadas a través de su antena. Estas ondas activan el microchip y a través de su antena transmite la información registrada en el microchip al lector. Para acabar el lector envía la información obtenida del tag a una base de datos donde previamente se ha registrado las características del producto.

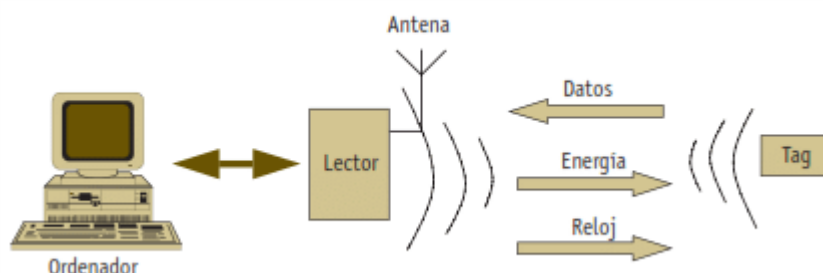


Figura 42 - Funcionamiento RFID

4.3.5 Sensor RFID en la Industria 4.0

En la actualidad la industria está avanzando al modelo industrial 4.0, conocidas como fábricas inteligentes, este modelo de fábrica permite mantener todos los objetos entre sí. En base a las explicaciones de Juan Manuel Ferrer⁷, el concepto IOT (Internet of Things), está cambiando rápidamente la forma de proceder. La famosa pirámide de la automatización (Modelo Industria 3.0) deja paso a una estructura donde todas las capas del modelo se comunican entre sí y los objetos y operadores humanos forman también parte del ecosistema productivo (Modelo Industria 4.0).

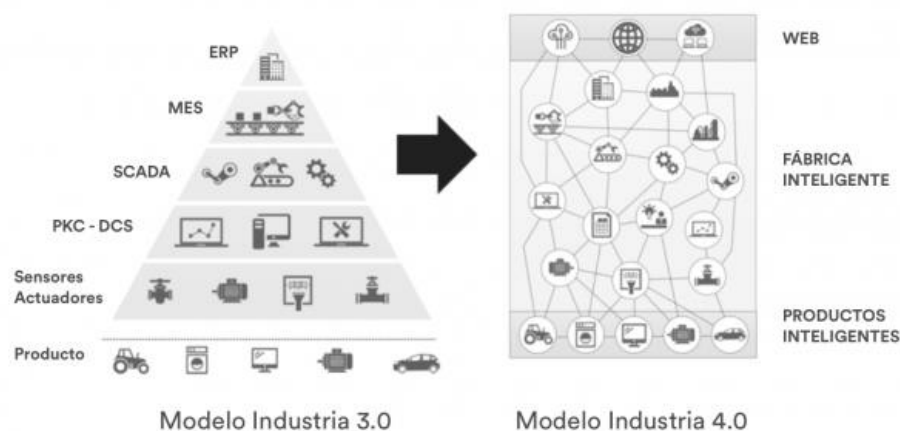


Figura 43 - Modelo Industria 3.0/ Modelo Industria 4.0

Los sensores RFID tienen un papel imprescindible en el avance a la industria 4.0, ya que los RFID habilitan la conexión de datos entre productos, sistemas de control, maquinaria y personas de forma autárquica, sencilla y barata. La transferencia de datos entre el producto y el sistema abre la puerta a multitud de opciones de personalización ya que es el propio producto el que instruye a la máquina en cuanto a lo que necesita en un determinado punto de la cadena de producción⁷.

En cuanto al suministro y logística del producto, las cualidades del RFID para el seguimiento y rastreo, permiten aumentar la visibilidad y mejorar la eficiencia de la cadena. También es imprescindible en las tareas de falsificación y facilita el paso de los productos en los controles fronterizos.

Con la información que contiene los identificadores puedes comprobar la autenticidad de los productos gracias a la información que almacenan. De esta manera los RFID extiende las capacidades de la industria 4.0 hasta el punto de venta.

5 Diseño del proceso de automatización

5.1 Aplicación

En el siguiente diagrama observamos los diferentes componentes de la celda industrial donde procederemos a realizar la simulación.

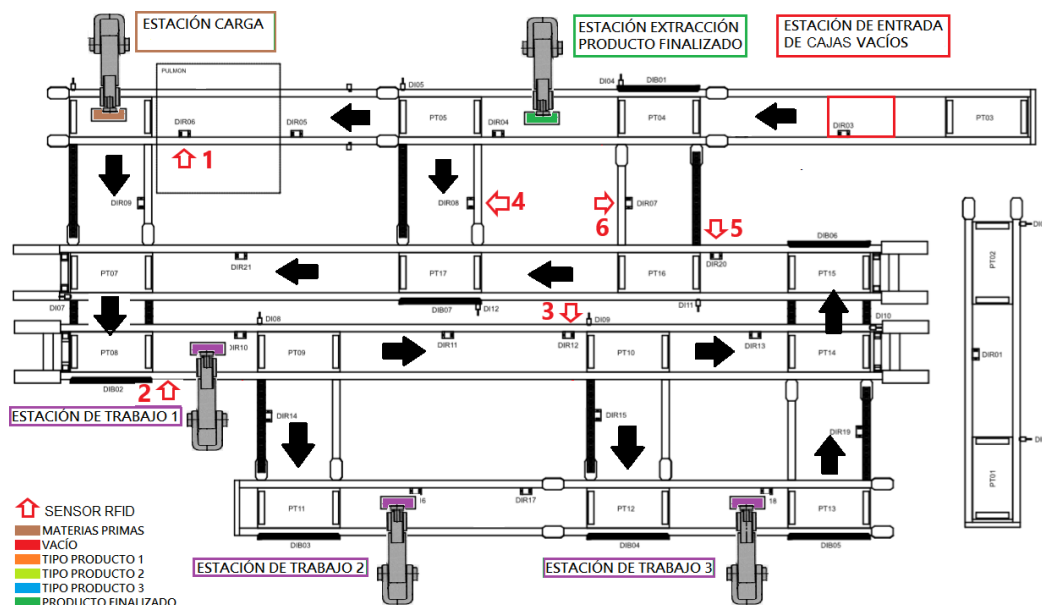


Figura 44 - Direcciones de movimiento de la celda

La dirección de movimiento de las cintas de la celda es unidireccional, esto quiere decir que solo se mueven en una dirección. El sentido de las cintas está indicado en la figura de arriba.

La simulación empieza con la entrada de cajas vacías preparadas para su posterior tratamiento, en la estación DIR03. Dichas cajas ya llevan pegado sus identificador que nos ayudara a saber que producto depositar en el paquete y que acabados deben realizarle a la caja.

Estos paquetes llegan a la bifurcación PT05 y si la estación de carga está libre (DIR05, DIR06 o PT06 en reposo) siguiera recto, en caso contrario tomara la otra dirección (PT17). Todos los paquetes que se redirijan a la plataforma 17 estará a la espera de la liberación de la estación de carga, hasta que no ocurra esto se mantendrá dando círculos. En la plataforma 16 se verificará si la estaciones DIR05, DIR06 o PT06 están reposo para redirigir las cajas a la zona de carga.

Antes de llegar a la plataforma 6, con la ayuda del sensor RFID 1, se reconocerá con qué tipo de producto se debe llenar dicho paquete, y se llevará a cabo el llenado del producto en la estación de carga. Una vez acabada la acción la caja se dirige a la estación de trabaja 1, donde se realizará algunos retoques a la caja. Todos los paquetes pasan por la estación de trabajo 1.

Antes de llegar a la estación de trabajo número 1, se ha identificado el paquete, con toda la información que nos proporciona el Tag decidiremos a donde dirigirlo. Si se trata de un producto de tipo 2 o 3 seguirá recto, en cambio, si es el producto 1 se dirigirá a la estación de trabajo número 2 donde será tratado y posteriormente se dirigirá a la plataforma 12, 13 y 14.

En cuanto a los productos 2 y 3 se dirigirán a la plataforma 10 donde se identificará que tipo de acción debe realizarse. Si el producto es tipo 2 seguirá recto hacia la plataforma 14 y si se trata de un producto tipo 3 se dirigirá a la plataforma 12 donde procederá a la estación de trabajo número 3 y posteriormente a las plataformas 13, 14.

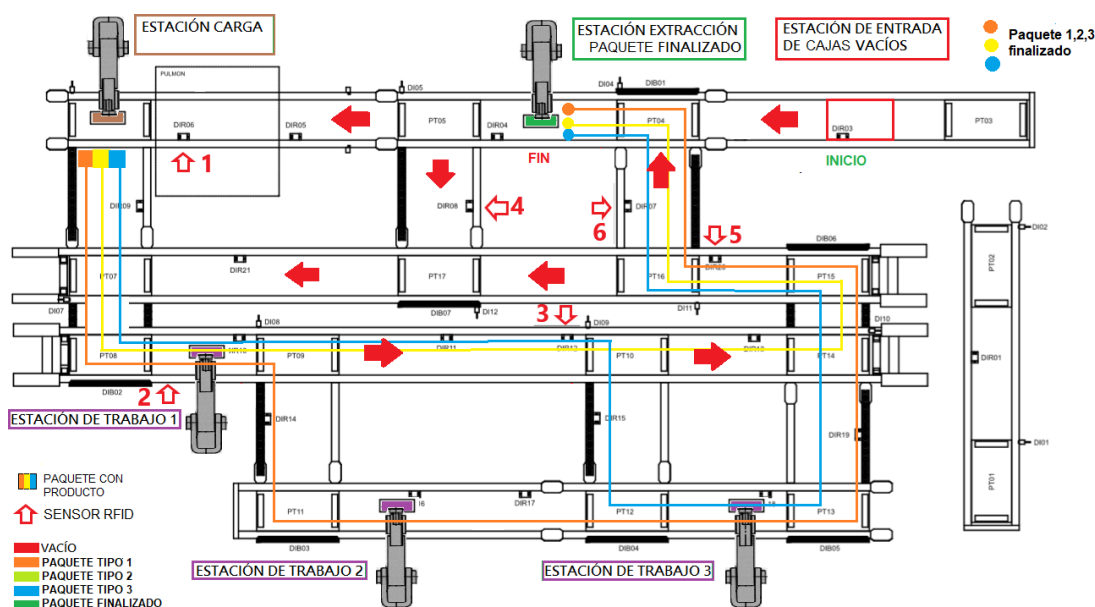


Figura 45 - Recorrido de los paquetes

También comentar que disponemos de 6 sensores RFID. El sensor 1,2, 3 y 5 nos ayudan a identificar qué tipo de paquete es para realizar una operación u otra, el sensor 4 y 6 son de control.

El sensor de reconocimiento número 1 identifica la caja vacía, posteriormente en la estación de carga rellenar dicha caja con el producto correspondiente. Como ya he dicho anteriormente las cajas vacías ya llevan un identificar que nos indica que producto debe colocarse en ella y los tratamientos que debe sufrir esta caja. El RFID número 2, 3 y 5 nos ayuda a redirigir los paquetes a una dirección determinada dependiente del tipo de paquete. El sensor número 4 nos informa de que tipo de paquete está a la espera de su llenado, ya que los paquetes que pasan por ese punto vienen vacíos porque la zona de carga está ocupada. Básicamente nos ayuda a saber qué tipo de producto no se ha llenado en ese momento. Y el 6 lleva un control de los paquetes que han cumplido todas las etapas de trabajo que les correspondía.

Se ha elaborado el diagrama de flujo del proceso ubicado en el documento

ANNEX_Diagrama_de_Flujo

5.2 Comunicaciones industriales

Las comunicaciones que se han llevado a cabo en nuestro proyecto son las que realizan los PLCs que se intercomunican entre ellos.

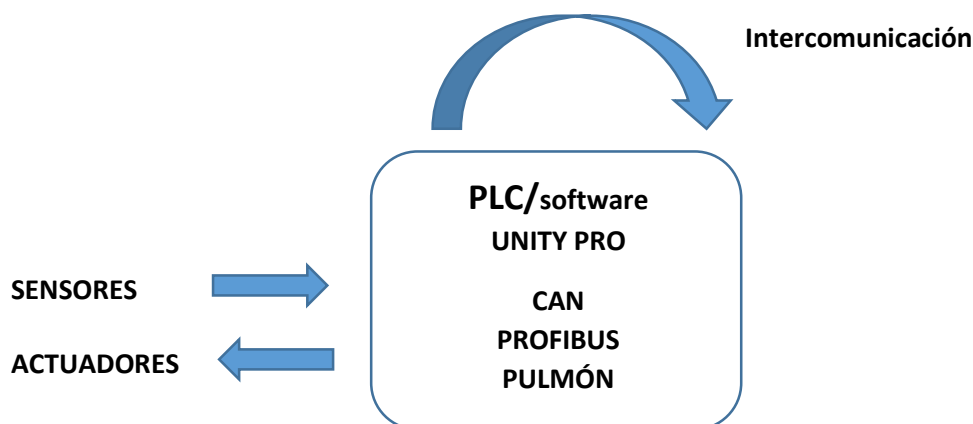


Figura 46 - Diagrama de comunicación

La conexión entre la salida y las entradas con los PLCs se realiza mediante un puerto serie y con los diferentes protocolos. El protocolo Profibus (Profibus DP) utiliza la capa física Rs485, CAN (Canopen), utiliza dos cables trenzados con una impedancia característica de $120\ \Omega$ y Ethernet utiliza un cableado RJ-45 en esta célula.

En nuestro proyecto las distintas líneas que forman la celda de trabajo (línea CAN, PROFIBUS Y PULMÓN) se intercambian información para el correcto funcionamiento del proceso. Ya que una vez llega la bandeja a al final de una línea, ha de transmitir y conocer el estado de los elementos de la siguiente línea.

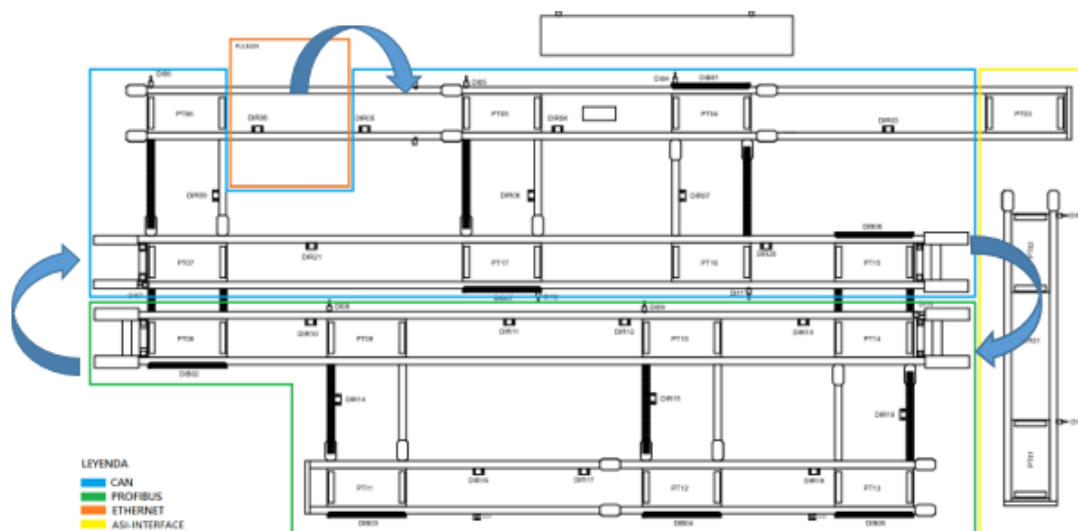


Figura 47 - Comunicación entre las líneas

5.3 Posición de los sensores de reconocimiento

Se han colocado los sensores RFID de manera simulada en nuestra celda de trabajo en función de la información que necesitamos que nos aporten en cada sitio. Hay sensores que se han situado antes de las estaciones de trabajo, otros antes de las intersecciones y otros básicamente para aportar información. A continuación, justificaremos la posición de cada uno de los sensores de nuestra celda de trabajo.

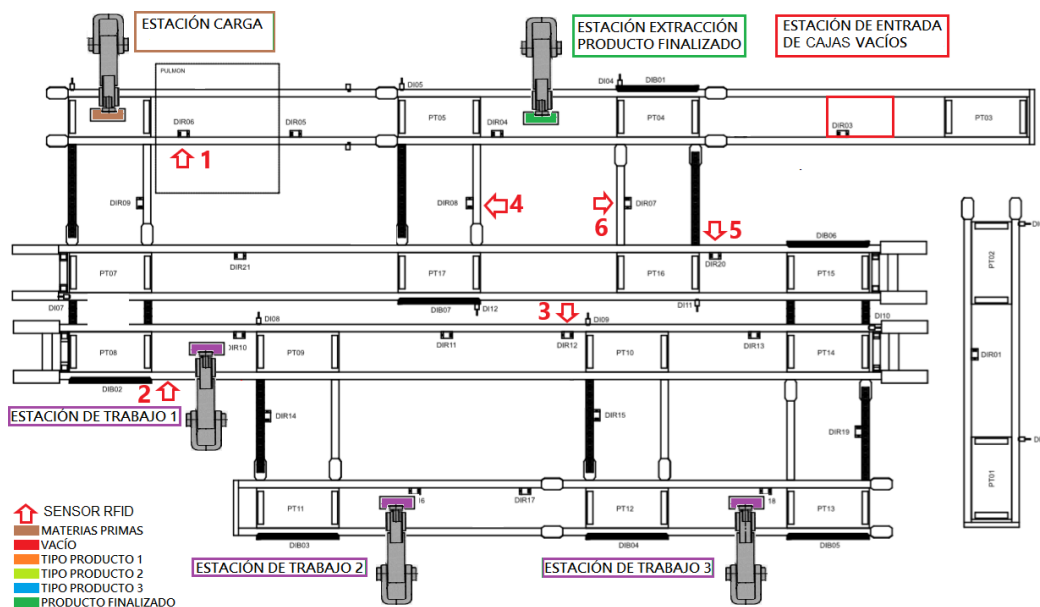


Figura 48 - Posición de los sensores

- Sensor 1: El RFID 1, se ha situado justo antes de la estación de carga (DIR06), ya que tenemos que saber con qué producto llenar el paquete que llega. Por lo tanto, hemos decidido situar el sensor 1 justo antes para identificar el paquete y de esta manera saber con qué producto debe ser rellenado. Finalmente, esta información se enviará a la estación de carga que una vez situada el paquete en la plataforma realizará la función de llenado.
- Sensor 2: El RFID 2, se ha situado justo antes de la estación de trabajo 1. Se ha situado en esa posición, en primer lugar, para informar a la estación de trabajo 1, si se trata de un paquete vacío o no. En segundo lugar, para direccionar el paquete a la posición correcta, ya que justo después de la estación de trabajo nos encontramos con una bifurcación.
- Sensor 3: El RFID 3, lo hemos colocado antes de la bifurcación (plataforma 10) (DIR12), básicamente está situado en esa posición para redirigir el paquete a la dirección correcta.
- Sensor 4: El identificador 4, lo hemos situado después de la bifurcación que se encuentra en la plataforma 6 (DIR08), ya que los paquetes que cogen esa dirección están vacíos y nos interesa identificarlos para saber que cajas no han sido llenadas, para posteriormente poder gestionarlas como cajas vacías.

- Sensor 5: también lo hemos situado antes de una bifurcación (plataforma 16) (DIR20), básicamente para gestionar si se trata de un paquete sin acabar, es decir, un paquete vacío, o de un paquete que ya ha completado todas las fases de tratamiento.
- Sensor 6: El RFID 6, se ha situado antes de la plataforma 4 (DIR07), para ser capaces de reconocer los paquetes que deben ser retirados en la estación de extracción. De no ser así seguirán dentro de la línea para que cumplan todos los acabados.

Como se puede observar en la figura 48, los sensores simulados se han intentado colocar junto a sensores reales de la celda de trabajo. Se ha realizado de esta manera para darle un toque más real. Ya que una vez la bandeja pase por el sensor real de la celda, también activara la lectura de nuestro sensor simulado.

5.4 Retenedores y plataformas

El funcionamiento principal de la línea está basado en el estado de los retenedores y plataformas. Son los elementos principales encargados de la circulación de los paquetes. Estos dispositivos pueden llegar a tener 5 combinaciones de 3 estados posibles. El estado reposo (REST en el código de programación) se activa (1) cuando no hay ninguna bandeja (paquete) en el elemento. El estado avance (MOVE), se activa (1) si está circulando la bandeja (paquete). La activación (1) del estado petición (READY) se produce cuando se detecta una bandeja (paquete) en el elemento de la línea.

Posibles combinaciones:

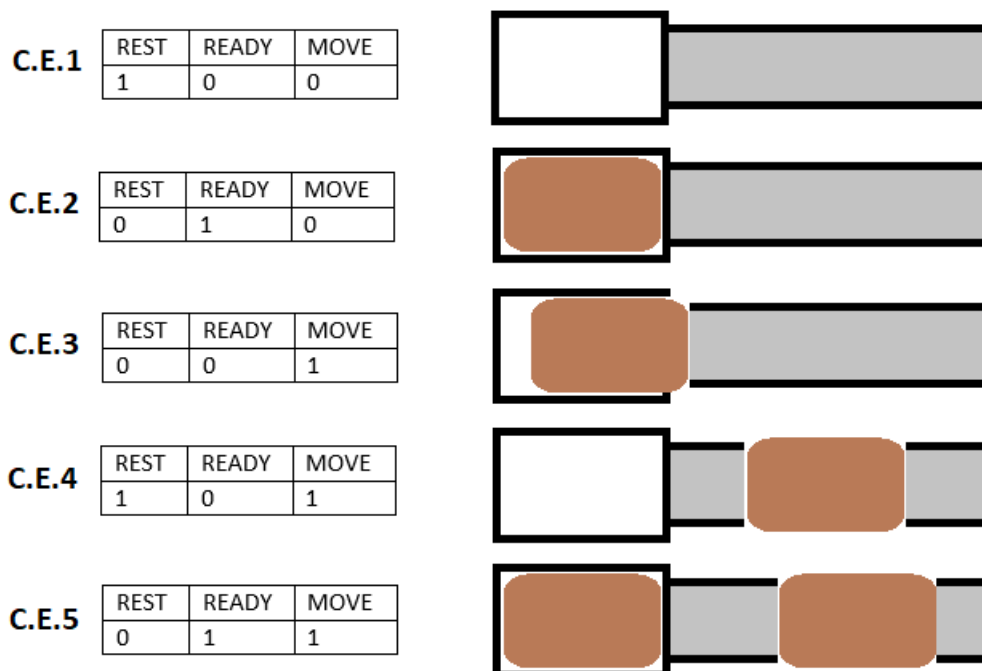


Figura 49 - Combinaciones Plataformas/Retenedores

C.E.1: Plataforma o retenedor en reposo.

C.E.2: Plataforma o retenedor detecta paquete.

C.E.3: Paquete en movimiento, pero la plataforma o retenedor aún no está libre para pasar a reposo.

C.E.4: Paquete en movimiento y además el elemento queda libre, por lo tanto, se activa el estado de reposo de la plataforma o retenedor.

C.E.5: Paquete en movimiento MOVE (1), además ha llegado otro paquete al elemento REST (0), pero el movimiento de este paquete, aunque este READY (1) no se ejecutara hasta que el paquete que está en movimiento llegue a su destino.

La circulación de los paquetes depende de la programación de la combinación explicadas anteriormente, la programación de estos elementos la podemos encontrar en las secciones LINEA y SALIDAS.

Comentar que partimos con la estación de trabajo teniendo en cuenta que no hay ningún paquete en la celda, por lo tanto, inicializamos todas las variables en reposo. REST (1), MOVE (0), READY (0).

La sección SETANDEREST, es la encargada de inicializar todas las variables teniendo en cuenta que en la celda de trabajo no hay ningún paquete.

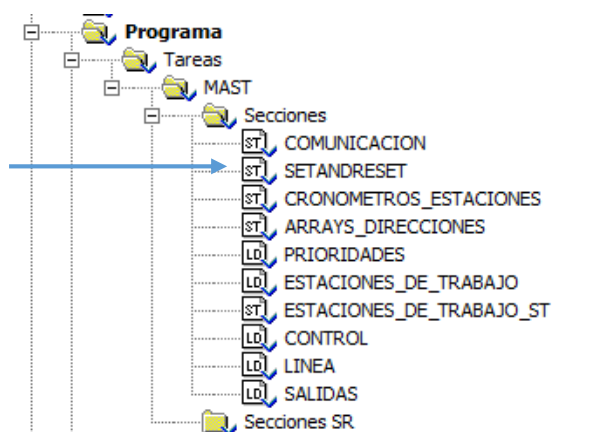


Figura 51 - Sección SETANDRESET

```
DIR03_REST := TRUE;
DIR03_READY := FALSE;
DIR03_MOVE := FALSE;
```

Figura 50 - Inicialización Variables

Partiendo que todas las variables están inicializadas, una vez llegue un paquete a un elemento se dará lugar a la activación del estado de petición y se desactivara es estado de reposo.

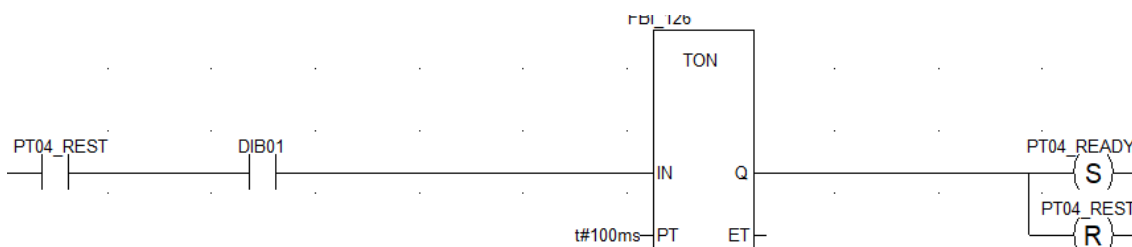


Figura 52 - Línea Ladder activación Ready

Como podemos observar en figura 52, una vez detecte el sensor DIB01 la llegada de un paquete y la plataforma PT04 este en reposo se activará el estado de petición y se desactivará el reposo de la plataforma 04. El temporizador TON se ha añadido para evitar error de falsas detecciones. Se trata de un sistema para filtrar la señal y asegurarnos que realmente el sensor ha detectado un paquete. El sensor DI05 está direccionado a un sensor real.

Nombre	Valor
Nombre	DIB01
Comentario	
Dirección	%IW\3.3\0.0.0.167.5
Programa RW	<input type="checkbox"/>
Valor	
Constante	<input type="checkbox"/>
Guardar	<input type="checkbox"/>
Datos globales	NO
Tipo	BOOL

Figura 54 - Direccionamiento del sensor

Una vez activado el estado de petición, la siguiente estación este en estado reposo y no haya ningún otro paquete en movimiento hacia el mismo destino se procederá a activar el estado de avance.

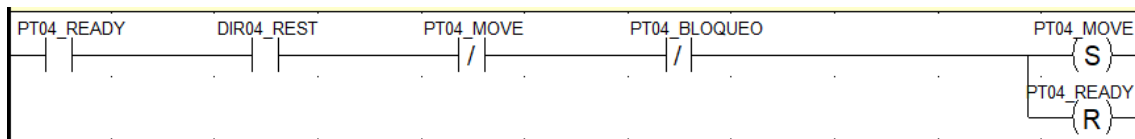


Figura 53 - Línea Ladder activación Move

La celda de trabajo está formada por varias líneas que se intercomunican entre ellas, debemos informar a la otra línea del estado de los elementos que están en frontera para la correcta circulación de los paquetes.

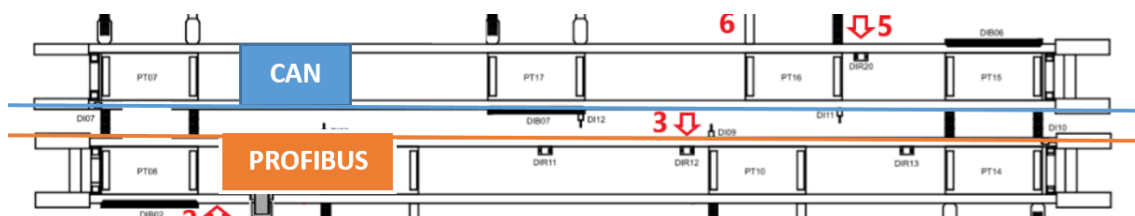


Figura 55 - Línea CAN y Profibus

El estado reposo de la plataforma PT15_REST (%MW303.0) es leído por la línea Profibus de la línea CAN. El estado reposo de la estación PT08_REST (%MW350.0), escrito por la línea Profibus a la línea CAN.

COMUNICACIÓN	
PROFIBUS	CAN
<pre>%MW350.0 := PT08_REST;</pre> <pre>PT15_REST := %MW303.0;</pre>	<pre>%MW303.0 := PT15_REST;</pre> <pre>PT08_REST := %MW350.0;</pre>

Figura 56 - Sección comunicación PT08/PT15

Intersección: 1 entrada, 2 salidas

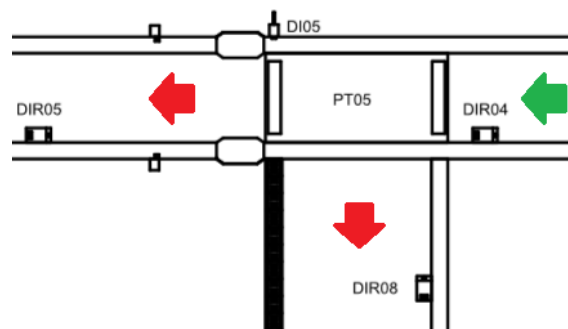


Figura 57 - Intersección 1 entrada/2 salidas

La programación para gestionar la dirección que debe tomar el paquete es una estructura similar a la de los casos anteriores. En este caso añadimos una línea de código para cada uno de los desvíos. El paquete se dirigirá a la dirección que activemos siempre y cuando se activen las otras variables.

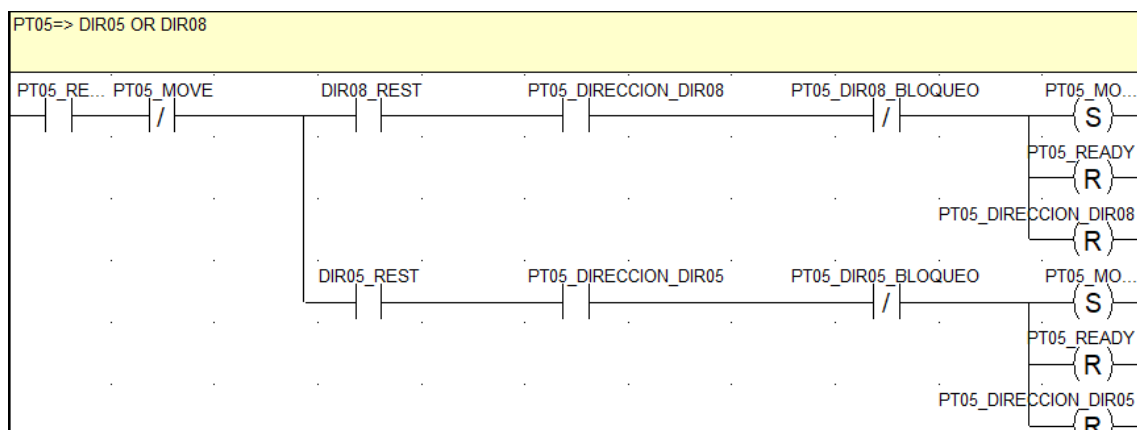


Figura 58 - Línea Ladder Activación Move 2 direcciones

En la figura 58, observamos que una vez seleccionado una dirección la otra es bloqueada por seguridad.

Intersección: 2 entrada, 1 salidas

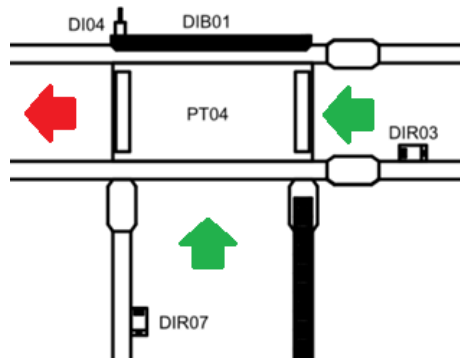


Figura 59 - Intersección 2 entrada/1 salidas

En este caso hemos tenido en cuenta que no se puede producir a la vez dos movimientos de entrada a una plataforma. Para asegurarnos que no sucede se bloquea el movimiento de una de la entras si la otra ya está en movimiento.

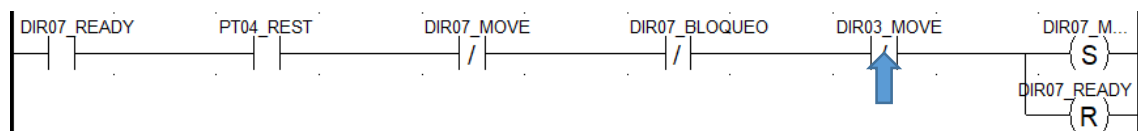


Figura 60 - Línea Ladder activación Move intersección 2 entrada/1 salidas

Si las dos entradas están en petición a la espera de entrar a la plataforma. La entrada de los paquetes se realizará de manera intercalada.

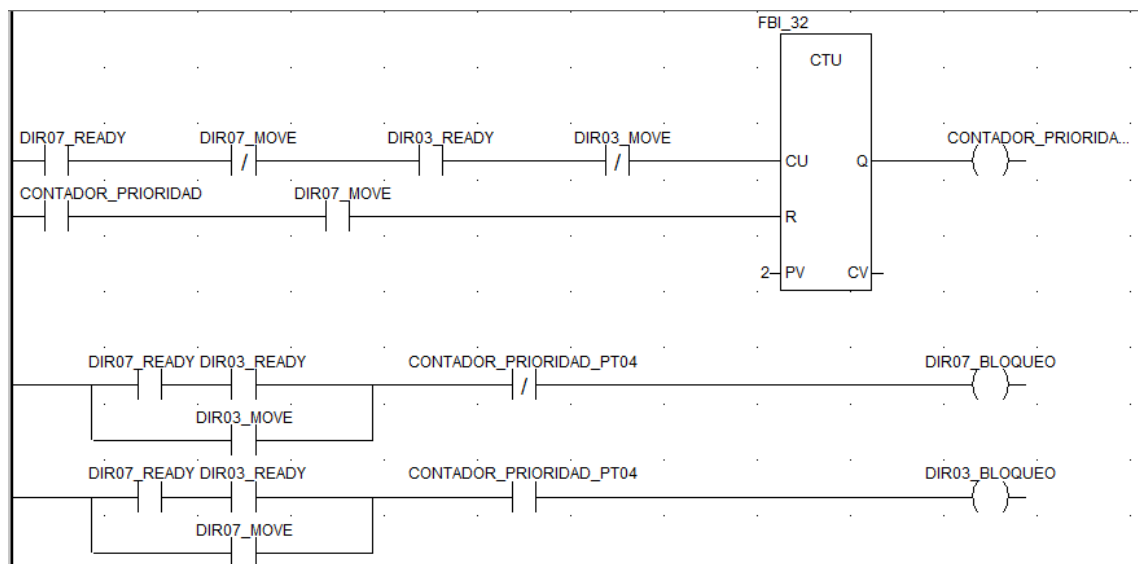


Figura 61 - - Línea Ladder prioridad

En algunos casos la combinación de los estados 4 y 5 nos posible por el tipo de programación. Esto sucede cuando la activación del estado reposo y la desactivación del estado avance se produce cuando el sensor de la estación destino detecta la llegada de un paquete. Es decir, que la estación origen solo estará lista para recibir otro paquete cuando la estación destino detecte el paquete que estaba en movimiento.



Figura 62 - Línea Ladder desactivación Move sin sensor retenedor

En cambio, en otros casos se permite la combinación de los estados 4 y 5, ya que la desactivación del estado de avance se produce cuando el paquete llega a la estación destino, pero el estado reposo de la estación de origen se activará justo después de detectar el paquete, por lo tanto, es posible la llegada de otro paquete.



Figura 63 - Línea Ladder desactivación Move con sensor retenedor

La señal de la salida del retenedor depende de la activación del estado de avance, la desactivación del estado de reposo y petición. Si se cumplen estas condiciones el pistón estará bajado y para asegurarnos que seguirá bajado hasta que pase el paquete se ha añadido un temporizador TOF.

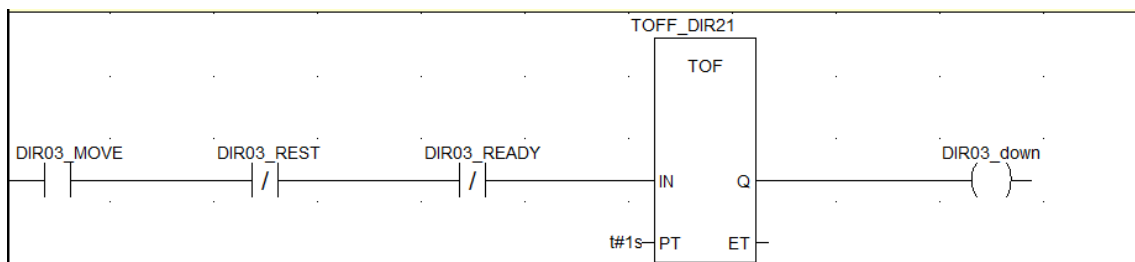


Figura 64 - Línea Ladder salida retenedor

El temporizador TOF se encarga de retener el pistón abajo durante el tiempo que le asignemos una vez cambien el estado de alguna de las condiciones anteriores.

La función de un temporizador TOF es realizar un retraso a la desconexión.

El retenedor DIR06 está aislado en la línea pulmón, por lo tanto, debemos informar a la línea Can de su estado. Se enviará y leerá información de su estado.

COMUNICACIÓN	
PULMÓN	CAN
<pre>%MW46.0 := DIR06;</pre> <pre>DIR06_DOWN := %MW10.0;</pre>	<pre>DIR06 := %MW46.0;</pre> <pre>%MW10.0 := DIR06_DOWN;</pre>

Figura 65 - Sección comunicación DIR06

5.5 Sensores RFID

5.5.1 Activación y lectura

La celda industrial está formada por seis sensores RFID, el proceso de lectura del identificador se simulará. A continuación, se explicará cómo se ha realizado la simulación.

La activación (entrada) de los sensores de reconocimiento se realiza aprovechando sensores reales de la celda de trabajo, en este caso hemos utilizado los sensores encargados de detectar la llegada de una bandeja, así una vez llega la bandeja a uno de los sensores, se activaría la lectura de nuestro sensor.

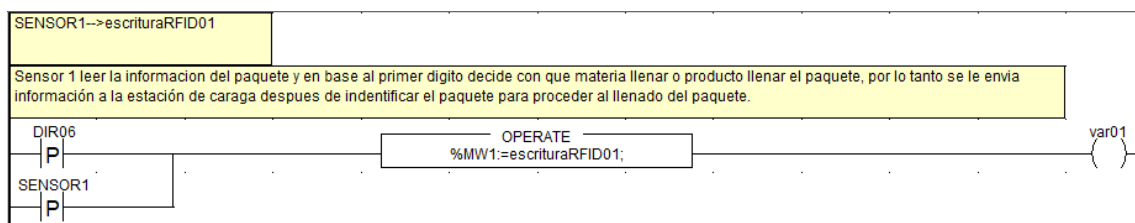


Figura 66 - Línea Ladder simulación lectura RFID

Como podemos observar en la figura 66, una vez detectado un flanco ascendente del sensor DIR06 de la celda, se procederá a registrar en una variable el contenido del identificador. También lo podemos realizar de manera manual con la una variable (SENSOR1) que hemos habilitado.

En cuanto a número de identificación que posteriormente leerá el RFID, se ha introducido manualmente desde la pantalla del operador.

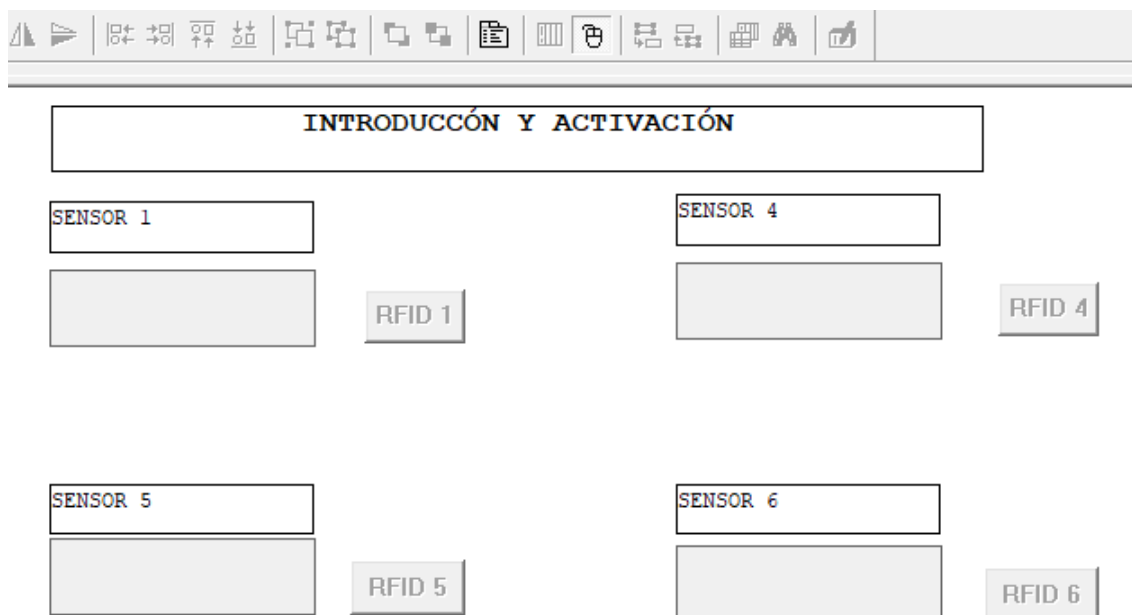


Figura 67 - Pantalla operador Introducción/Activación RFID

La simulación consiste en introducir de manera manual un dígito que se registrara en una variable. Posteriormente cuando se active la entrada del RFID, se copiará en una variable el dígito introducido anteriormente en la pantalla del operador para ser tratado.

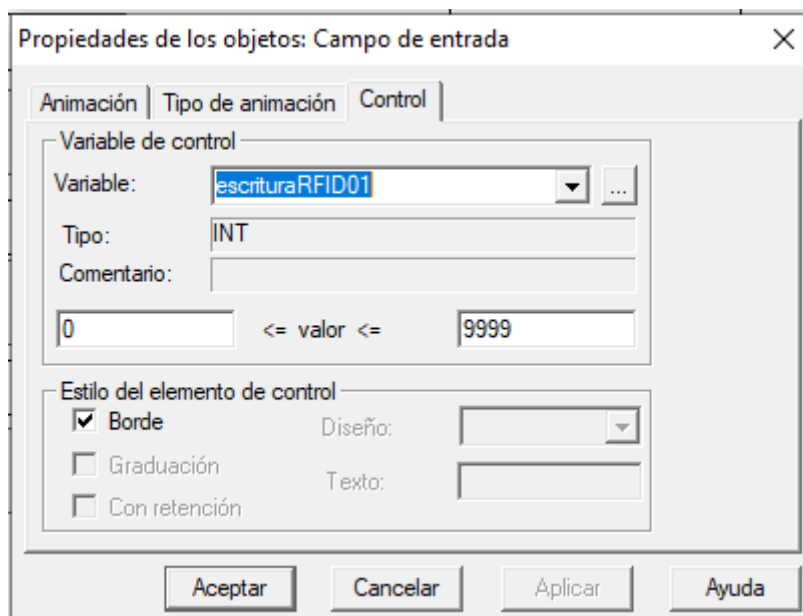


Figura 68 – Ejemplo Campo de entrada (escrituraRFID01)

Ejemplo:

En la pantalla del operador introducimos el dígito deseado (0-9999).

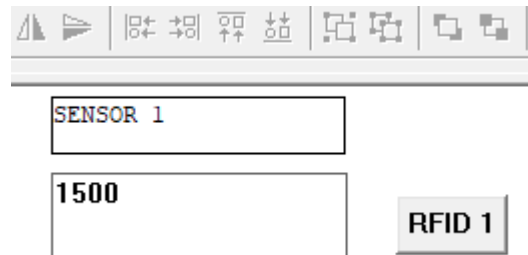


Figura 69 - Ejemplo introducción de dígito sensor 1

Este valor introducido es guardado en la variable **escrituraRFID01**, que simulara la etiqueta del paquete.

escrituraRFID01	1500	INT	Valor obtenido del RFID01(simulado)
-----------------	------	-----	-------------------------------------

Figura 70 - Ejemplo Tabla de animación (escrituraRFID01)

Ahora una vez detecte el sensor DIR06 la llegada de una bandeja o activemos de manera manual la variable SENSOR1, se simulará la lectura del paquete. Por lo tanto, se copiará en otra variable (%MW1) el valor introducido en la variable **escrituraRFID01**.

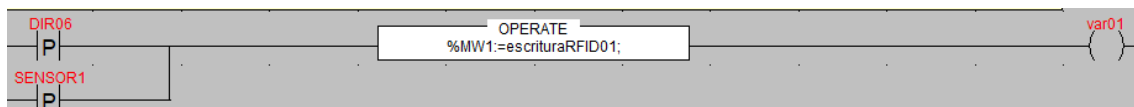


Figura 71 - Ejemplo simulación de lectura (escrituraRFID01)

Esta es la manera que hemos utilizado para simular nuestro proceso de activación, lectura y registro.

5.5.2 Identificación del paquete y tratamiento del dígito

Como ya he comentado anteriormente nuestros paquetes vienen un con identificador, con la ayuda del sensor RFID vamos a ser capaces de leer la información que contiene. Los

0123

Figura 72 - Ejemplo dígitos

identificadores de los paquetes están formados por 4 dígitos (0-9999). Cada dígito nos proporciona una información en concreto, es decir que se tratara cada dígito de manera individual. Este mecanismo nos ayudara a gestionar mejor la línea, nos proporciona mayor flexibilidad a la hora de aplicar modificaciones.

La línea de producción actualmente dispone de 3 productos distintos, y cada uno de ellos tiene unos acabados distintos. Pero puede que en un futuro haya que modificar el acabado de algún producto. Al haber dividido de manera independiente cada dígito no será necesario modificar ninguna parte del código del PLC, solo bastaría con modificar la parte correspondiente del identificador del paquete.

En el proceso actual solo se usará los 3 últimos dígitos, ya que es suficiente para abarcar las necesidades de la línea.




	Tipo de producto Producto 1 → 0-3 Producto 2 → 4-6 Producto 3 → 7-9
	Informa si el paquete debe ser tratado por la estación de trabajo número 2. Ser tratado → 0-4 No ser tratado → 5-9
	Informa si el paquete debe ser tratado por la estación de trabajo número 3. Ser tratado → 0-4 No ser tratado → 5-9

Tabla 3 - Clasificación de los dígitos

¿Cómo tratar cada dígito de manera independiente?

Tratamiento de los datos obtenidos por el sensor

Una vez registrado el dígito en el PLC procedemos a analizar cada dígito de manera independiente en función de la posición del paquete en la línea. Este tipo de tratamiento se realizará cuando el paquete llegue al RFID 1,2 o 3.

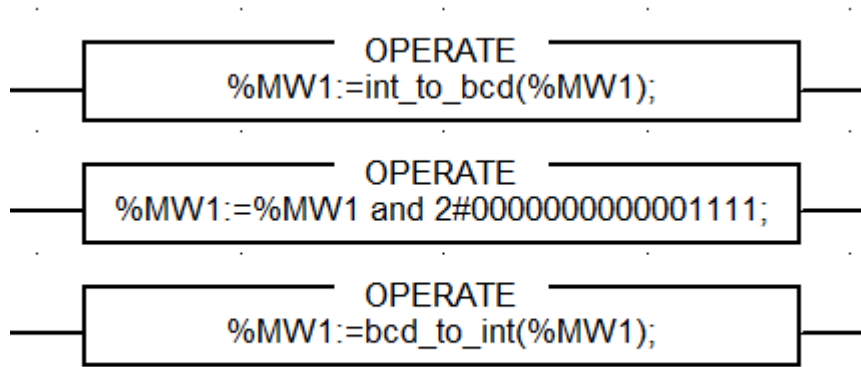


Figura 73 - Ejemplo tratamientos del último dígito

Último dígito

0123

Figura 74 - Último dígito

En cuanto al tratamiento del último dígito, que nos informa del tipo de producto. Primero de todo transformamos la variable de INT (16 bits) a BCD, de esta manera tenemos la representación de cada dígito de manera independiente. A continuación, con la ayuda de la función lógica AND eliminamos los dígitos que no nos interesan bit a bit. Finalmente transformamos de nuevo la variable de BCD a INT (16 bits).

Ejemplo:

%MW1: 0198

Representación in binario → 2#0000 0001 0010 1010 **Binario natural**

INT TO BCD → 2# 0000 0001 1001 1000 **binario BCD**

%MW1 AND 2#0000 0000 0000 1111

	0000 0001 1001 1000
AND	0000 0000 0000 1111
	<hr/>
	0000 0000 0000 1000

Figura 75 - Ejemplo operación AND (Último dígito)

BCD TO INT → **%MW1 = 8**

Después de realizar estos pasos obtenemos solo el último dígito.

Penúltimos dígito

0123

Figura 76 - Penúltimos dígito

El penúltimo dígito informa si un paquete debe ser tratado por la estación de trabajo 2. Para obtener únicamente el penúltimo dígito procederemos a realizar los siguientes pasos.

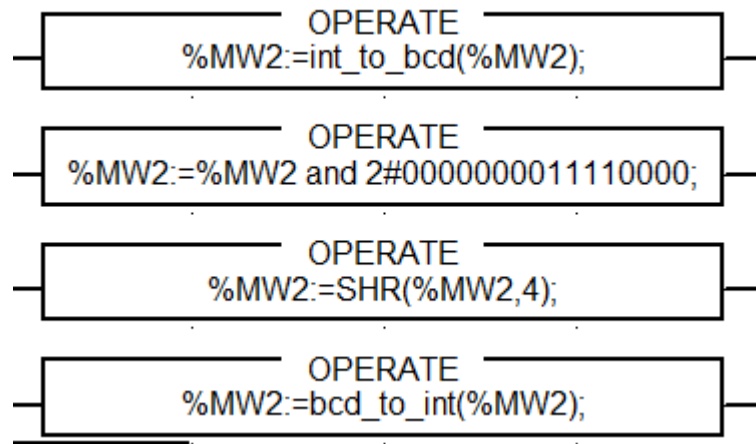


Figura 77 - Ejemplo tratamientos del penúltimo dígito

Como en el ejemplo anterior procederemos a transformar la variable X de tipo INT (16bit) a una variable de tipo BCD, posteriormente se eliminamos los dígitos que deseamos con la función lógica AND. Pero en este caso el dígito de interés no se encuentra en la última posición, es decir que si lo convertimos directamente a una variable INT obtendremos un valor erróneo. Por lo tanto, antes de proceder a convertir de nuevo la variable debemos desplazar los bits, en este caso 4 posiciones con la ayuda de la función SHR. Y finalmente proceder a convertir la variable.

Ejemplo:

%MW2: 0198

Representación in binario → %MW2: 2#0000 0001 0010 1010 **Binario natural**

INT TO BCD → %MW2: 2# 0000 0001 1001 1000 **binario BCD**

%MW2 AND 2#0000 0000 1111 0000

	0000 0001 1001 1000
	0000 0000 1111 0000
AND	<hr/>
	0000 0000 1001 0000

Figura 78 - Ejemplo operación AND (penúltimo dígito)

SHR(%MW2,4) → desplazar los bits 4 posiciones



Figura 79 - Desplazar 4 bits

%MW2 = 9

Antepenúltimo dígito

0123

Figura 80 - Antepenúltimo dígito

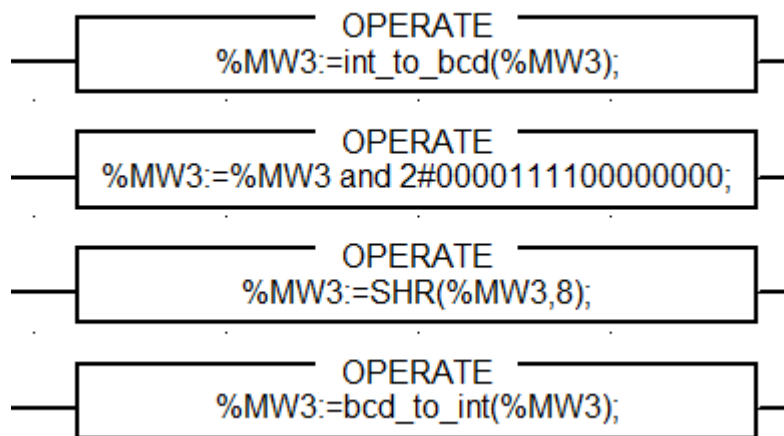


Figura 81 - Ejemplo tratamientos del antepenúltimo dígito

El segundo dígito o antepenúltimo informa si la caja debe ser tratada en la estación 3. Para obtener dicho dígito y poder analizarlo de manera individual realizaremos el mismo proceso que en los casos anteriores, pero en este caso se después de eliminar el valor de los dígitos de no deseados, se procederá a desplazar 8 posiciones los bits del número obtenido.

Ejemplo:

%MW3: 0198

Representación in binario → 2#0000 0001 0010 1010 **Binario natural**

INT TO BCD → 2# 0000 0001 1001 1000 **binario BCD**

%MW3 AND 2#0000 1111 0000 0000

$$\begin{array}{r}
 \text{AND} \quad \begin{array}{r} 0000 \ 0001 \ 1001 \ 1000 \\ 0000 \ 1111 \ 0000 \ 0000 \\ \hline 0000 \ 0001 \ 0000 \ 0000 \end{array}
 \end{array}$$

Figura 82 - Ejemplo operación AND (antepenúltimo)

SHR(%MW3,8) → desplazar los bits 8 posiciones



Figura 83 - Desplazar 8 bits

%MW3 = 1

5.5.3 RFID 1,2,3

Una vez detectado el identificado del paquete y registrado en una variable (del PLC) el dígito que contiene el Tag. Además, en el caso de los RFID 1,2 y 3 se secciona únicamente el dígito de interés (aplicando el tratamiento de datos explicado anteriormente) ya que es el que contiene la información necesaria para ese punto en concreto.

Una vez realizados todos esos pasos, se ha optado por la función **COMPARE** de Unity PRO, que nos comparará el dígito de interés con los rangos que hemos predeterminado para diferenciar los productos y las direcciones que debe tomar el paquete.

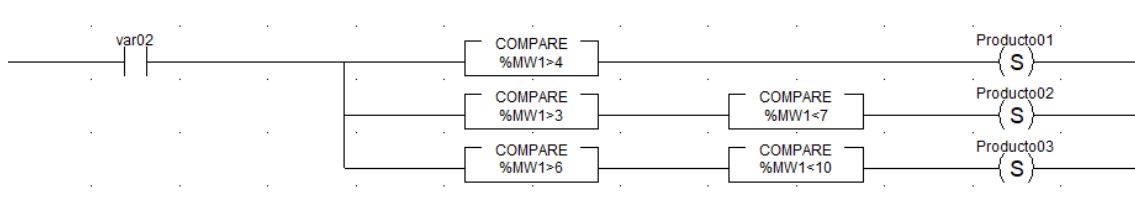


Figura 84 - Línea Ladder selección de producto

En la figura 84, observamos el uso de los COMPARE para determinar con qué producto llenar el paquete.

En el caso del RFID 2 y 3 aparte de dirigir el paquete a una dirección en función del rango donde se encuentre el dígito, primero de todo verificamos si no se trata de un paquete vacío.

La información de los paquetes vacíos nos la proporciona el RFID 4 de la línea CAN, y como el RFID 2 y 3, se encuentran en la línea PROFIBUS debemos establecer una comunicación para informar a esa línea de los paquetes vacíos. En este caso la línea PROFIBUS lee los registros (registro1(%MW335), registro2(%MW336), registro3(%MW337)).

COMUNICACIÓN	
CAN	PROFIBUS
<code>%MW335 := registro1;</code>	<code>registro1 := %MW335;</code>
<code>%MW336 := registro2;</code>	<code>registro2 := %MW336;</code>
<code>%MW337 := registro3;</code>	<code>registro3 := %MW337;</code>

Figura 85 - Sección comunicación registros

Con la información recibida de la línea CAN verificamos si el paquete identificado no está vacío.

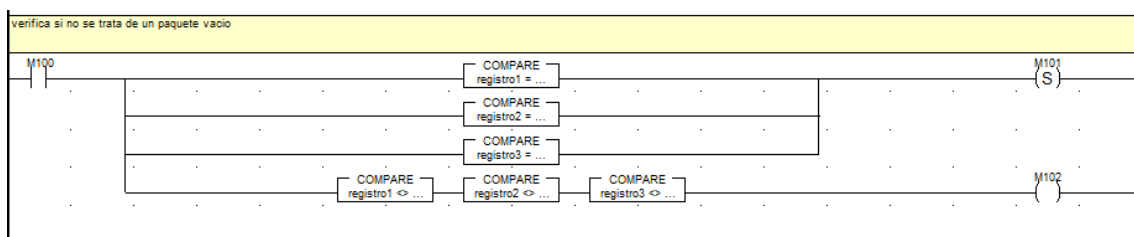


Figura 86 -- Línea Ladder activación M102

5.5.4 RFID 4,5,6

El sensor de reconocimiento 4 es el encargado de reconocer los paquetes vacíos. Una vez reconocidos guardamos los ID de estos paquetes en una serie de variable para tener constancia de ellos. A continuación, se mostrar el proceso de guardado:

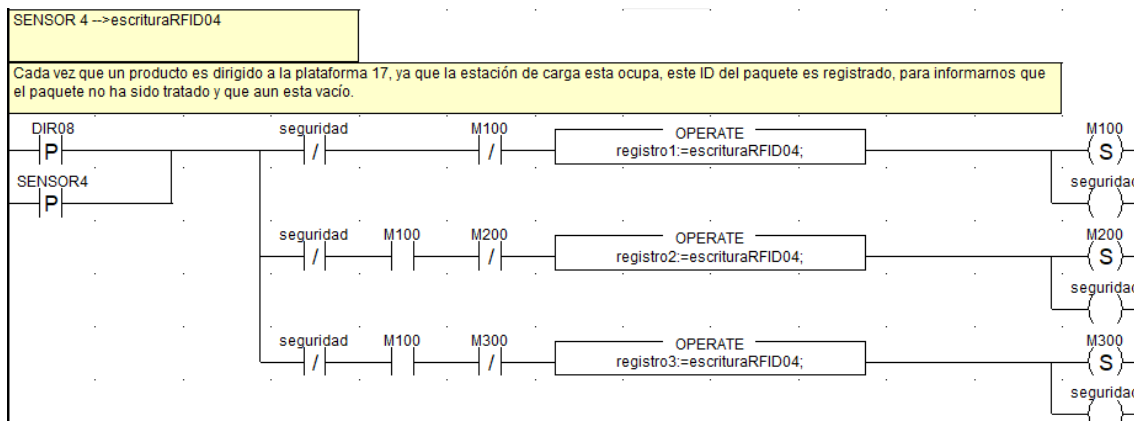


Figura 87 - Línea Ladder asignación de registros

En la figura 87, podemos ver como se intercala el guardado de datos, y solo se procederá a guardar el ID en la variable que este libre. En este caso solo podemos registrar tres paquetes vacíos, ya que hemos considerado que es más que suficiente. Si se diera el caso de necesitar más, bastaría con añadir más registros.

El caso de RFID 5 se encargará de diferenciar los paquetes vacíos de los paquetes finalizados.

Una vez diferenciado, si se trata de un paquete vacío, analizamos el estado de la estación de carga. Si está libre mandamos el paquete vacío para ser llenado, si está ocupada el paquete vacío seguirá a la estación de su libración (estación de carga).

```
(*modificado*)
(* observamos si se trata de un paquete vacío, si es así verificamos si la estación de carga está libre*)
IF (PT16_READY = TRUE) AND ((escrituraRFID05 = registro1) OR (escrituraRFID05 = registro2)
    OR (escrituraRFID05 = registro3)) THEN

    IF (PT06_REST = TRUE OR DIR06_REST = TRUE OR DIR05_REST = TRUE) THEN

        PT16_DIRECCION_DIR07 := TRUE;

    ELSE

        PT16_DIRECCION_PT17 := TRUE;
    END_IF;

END_IF;

(* verificamos que se trate de un paquete ya procesado para su posterior extracción *)
IF (PT16_READY = TRUE) AND ((escrituraRFID05 <> registro1) OR (escrituraRFID05 <> registro2)
    OR (escrituraRFID05 <> registro3)) THEN

    PT16_DIRECCION_DIR07 := TRUE;

END_IF;
```

Figura 88 - Línea ST activación (PT16_DIRECCION_DIR07/PT16_DIRECCION_DIR017)

El sensor 6 también nos proporciona información que gestionaremos para saber si se trata de un paquete vacío o de un paquete finalizado.

Si se trata de un paquete vacío borraremos ese ID de los registros ya que este paquete se dirigirá a la estación de carga. Y si se trata de un paquete finalizado enviaremos un aviso (set VAR_PRODUCTO_FIN) a la estación de extracción para que sea retirado.

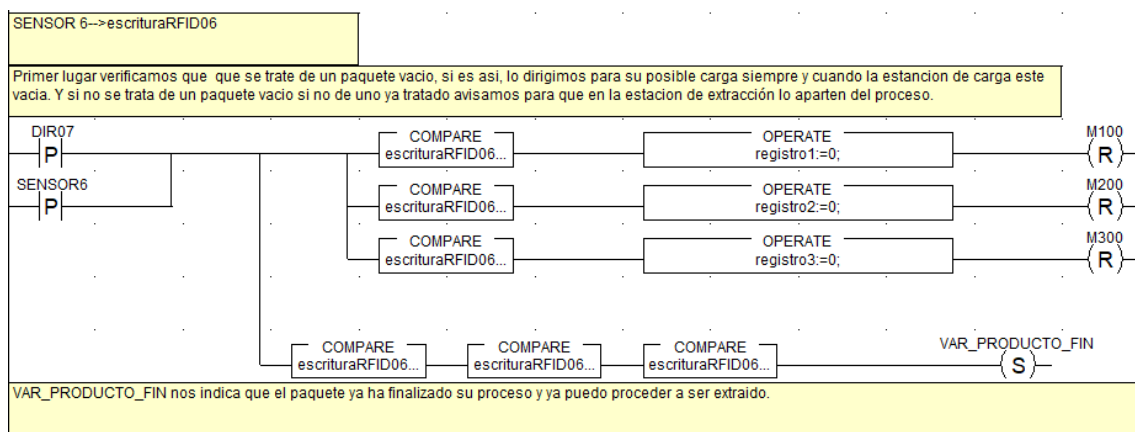


Figura 89 - Línea Ladder activación VAR_PRODUCTO_FIN/Reset (M100, M200, M300)

5.6 Motores

La celda está formada por una serie de motores que accionan las cintas de la línea industrial. Para la activación de los distintos motores y las plataformas se utilizan los estados que nos indican el avance del paquete (MOVE).

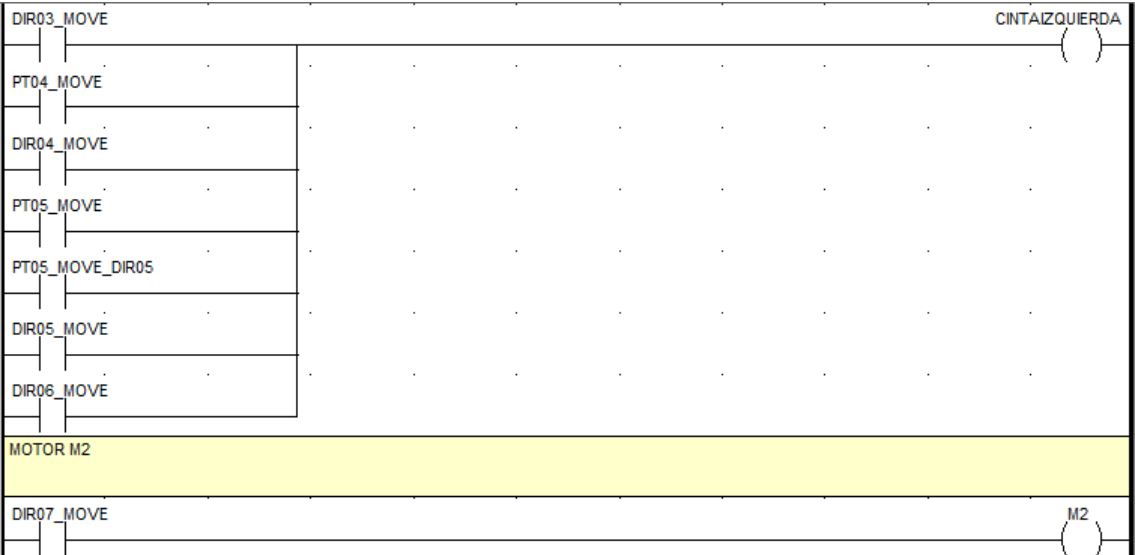


Figura 90 - Línea Ladder Motores

A continuación, se mostrar la posición de los motores en la celda industrial con los nombres utilizados a la programar.

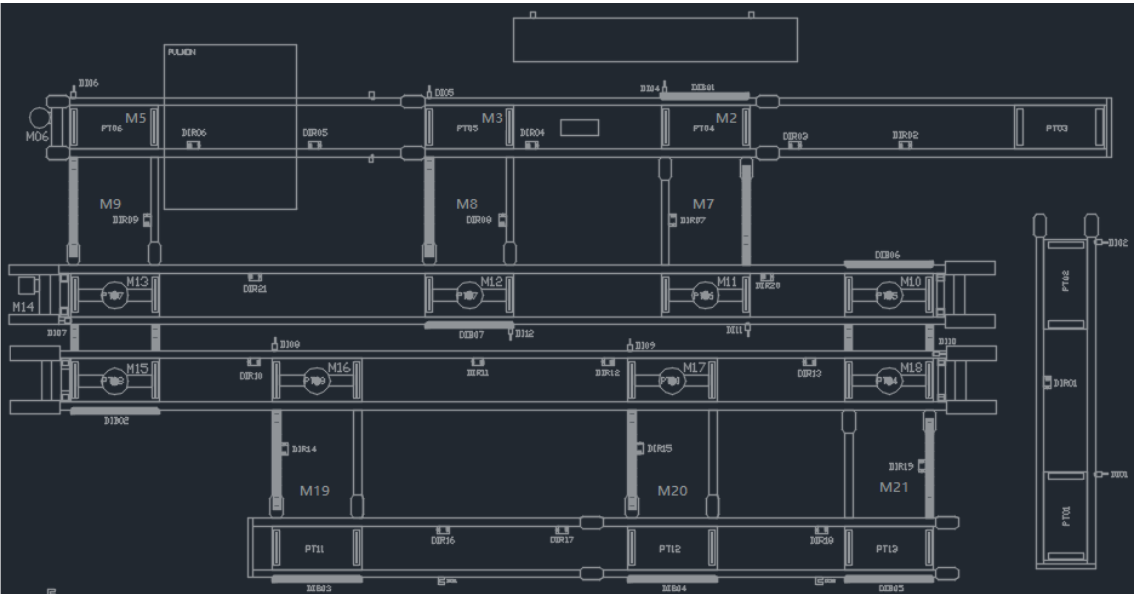


Figura 91 - Posición Motores

5.7 Trazabilidad

Para saber que está sucediendo en la línea en todo momento se ha elaborado una serie de medidas para llevar el control del proceso.

Los sensores RFID de la celda nos dan información de los paquetes. Por lo tanto, tendremos constancia en todo momento de la posición de nuestros paquetes dentro de la celda de trabajo.

Usando las Pantallas del operador podremos visualizar de una manera cómoda el paquete que está pasando por esa posición.

ID SENSORES		ID SENSORES	
	RFID 02		RFID 01
	RFID 03		RFID 04
			RFID 05
			RFID 06

Figura 92 - Pantalla del operador ID

Otra de las medidas para llevar un buen control de la celda de trabajo consiste en guardar en una Array los ID de los paquetes junto al tiempo que ha permanecido en cualquier estación de trabajo de nuestra celda.

Modificación Forzar		
Nombre	Valor	Tipo
ARRAY_TIEMPO_M_P		ARRAY[1..4] OF...
ARRAY_TIEMPO_M_P[1]		TIPO_TIEMPO
CRONO		TIME
ID		INT
+ ARRAY_TIEMPO_M_P[2]		TIPO_TIEMPO
+ ARRAY_TIEMPO_M_P[3]		TIPO_TIEMPO
+ ARRAY_TIEMPO_M_P[4]		TIPO_TIEMPO
ARRAY_TIEMPO_P_F		ARRAY[1..4] OF...
ARRAY_TIEMPO_P_F[1]		TIPO_TIEMPO
CRONO		TIME
ID		INT
+ ARRAY_TIEMPO_P_F[2]		TIPO_TIEMPO
+ ARRAY_TIEMPO_P_F[3]		TIPO_TIEMPO
+ ARRAY_TIEMPO_P_F[4]		TIPO_TIEMPO

Figura 93 - Array Tiempo

También podemos visualizar a tiempo real el tiempo que permanecen los paquetes en nuestras estaciones de trabajo.

TIEMPOS DE TRABAJO		
Tiempo Estación 1	Tiempo Estación 2	Tiempo Estación 3
TIME01	TIME02	TIME03

Figura 94 - Pantalla del operador (tiempos de trabajo)

Para finalizar nuestro proceso de rastreo y control, tenemos constancia de los paquetes vacíos que circulan por la celda de trabajo. Para ellos usamos variables para guardar su ID, de esta manera cada vez que pasen por un sensor de reconocimiento sabremos que se trata de un paquete vacío.

Nombre	Valor	Tipo	Comentario
registro1		INT	Paquete vacío
registro2		INT	Paquete vacío
registro3		INT	Paquete vacío

Figura 95 - Tabla de animación (registros)

Además, podemos visualizar a tiempo real los paquetes vacíos de la celda de trabajo con la Pantalla de operador.

Paquetes vacíos (REGISTOS)	
	REGISTRO 1
	REGISTRO 2
	REGISTRO 3

Figura 96 - Pantalla del operador (registros)

Con todas estas medidas conocemos la posición de nuestros paquetes, llevamos un registro de sus identificadores y podemos visualizar a tiempo real la información desde el inicio hasta el final del proceso.

5.8 Estaciones de trabajo

La celda de trabajo está formada por cinco estaciones de trabajo (simuladas) en las se podría colocar robots para realizar las funciones.

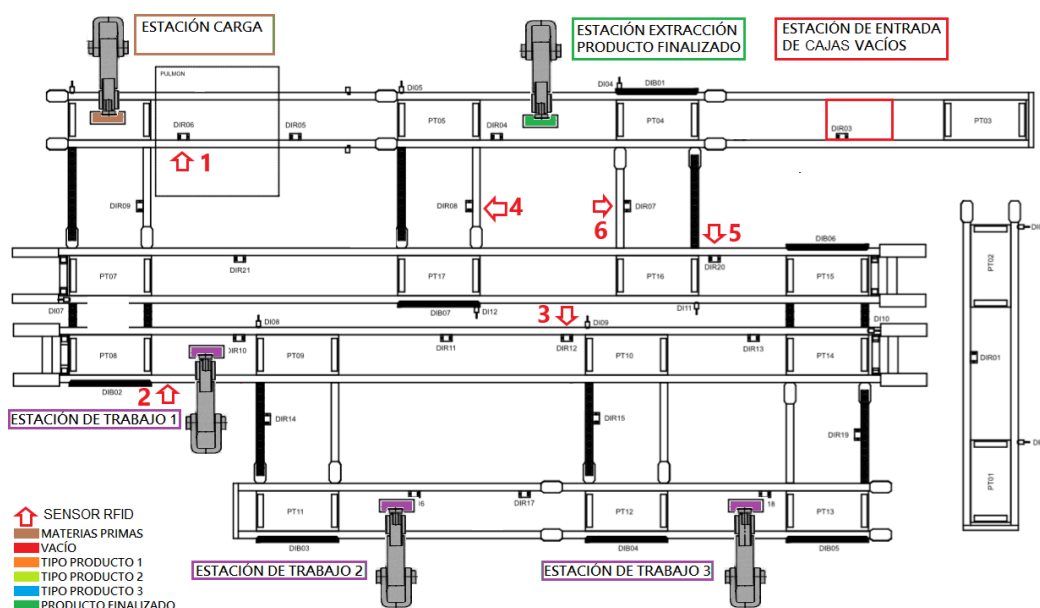


Figura 97 - Plano Celda de trabajo (Estaciones de trabajo)

La primera estación es la encargada de rellenar el paquete con su producto, siempre y cuando haya productos disponibles y el paquete este situado sobre la plataforma de carga (plataforma 6).

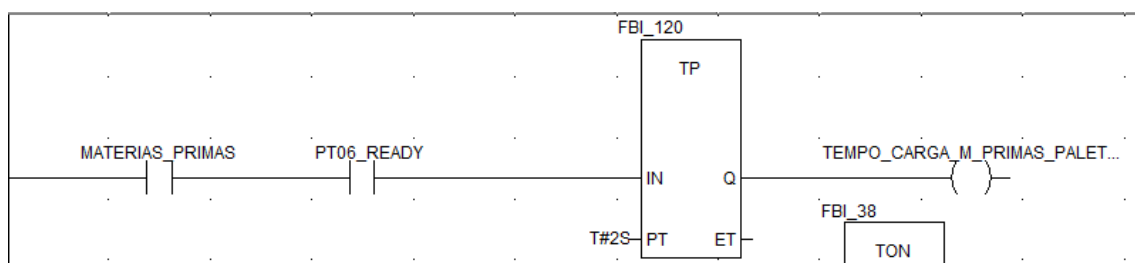


Figura 98 - Línea Ladder (estación de carga)

El temporizador TP simulara el tiempo de trabajo de la estación de carga. También se ha añadido un temporizador TON básicamente para informarnos del tiempo de ejecución de la tarea. Se guardará el tiempo transcurrido en una variable y se visualizará en la pantalla del operador.



Figura 99 - Línea Ladder (estación de carga) temporizador TON

Una vez se está realizando la acción de carga se bloqueará el movimiento del paquete hasta que haya finalizado. También se bloquea el movimiento si no hay productos disponibles para rellenar el paquete o si se está realizando un cambio de flujo.

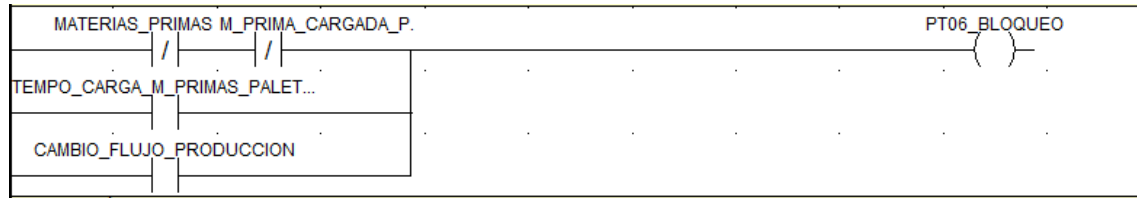


Figura 100 - Línea Ladder Bloque PT06

En la sección ESTACIONES_DE_TRABAJO_ST se ha tenido en cuenta el control del stock de materias. El siguiente código nos informa sobre la disponibilidad de productos en nuestro almacén.

```
(*INDICADOR DE LA EXISTENCIA DE MATERIAS PRIMAS EN LA LINEA*)
IF (CONTADOR_CANTIDAD_M_PRIMAS > 0) THEN
    MATERIAS_PRIMAS := TRUE;
ELSE
    MATERIAS_PRIMAS := FALSE;
END_IF;
```

Figura 101 - Línea ST (Existencia materias primas)

Una vez realizada la acción de rellenar el paquete (M_PRIMA_CARGADA_PT06) se procederá a reducir el contador de materias primas de nuestro Stock, si aún disponemos de productos.

```
(*DISMINUIR CANTIDAD MATERIAS PRIMAS DE LA LINEA*)
IF (RE (TEMPO_CARGA_M_PRIMAS_PALET_PT06)) = TRUE THEN
    M_PRIMA_CARGADA_PT06 := TRUE;
    IF (CONTADOR_CANTIDAD_M_PRIMAS > 0) THEN
        DEC (CONTADOR_CANTIDAD_M_PRIMAS);
    END_IF;
END_IF;
```

Figura 102 - Línea ST (Disminuir materias primas)

Se ha elaborado una tabla (ARRAY) que registrar el tiempo que se ha tardado en la estación. Guarda el tiempo y el ID del paquete para llevar un control de los tiempos de trabajo.

Una vez finalizado la acción de carga los paquetes se dirigen a las estaciones de trabajo 1,2 o 3, cada paquete realizara un recorrido en función de su ID. Como ya se ha explicado en el apartado “Sensores RFID”, los paquetes tomaran una u otra dirección según el rango en el que se encuentre el dígito analizado.

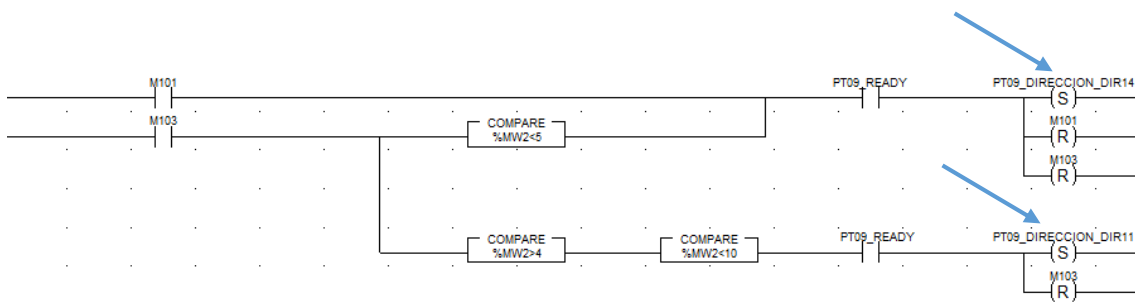


Figura 103 - Línea Ladder activación (PT09_DIRECCION_DIR14/PT09_DIRECCION_DIR11)

Las estaciones de trabajo 1,2 y 3, siguen la misma lógica de programación. Antes de permitir el acceso a la estación de trabajo reconocemos que el tipo de paquete para saber si se debe realizar la acción de manipulación o no.

```
(* SI DETECTO UN FLANCO ASCENDENTE EN LA PLATAFORMA 8 Y OTRO EN M102 ACTIVO LA ESTACION DE TRABAJO 1,
EL M102 ME INFORMA SI EL PAQUETE ES <0. *)
IF PT08_MOVE_F_A = TRUE AND M102_F_A = TRUE THEN
    TRABAJO_ESTACION_1 := TRUE;
```

Figura 104 - Línea ST activación (TRABAJO_ESTACION_1)

En el caso de la estación de trabajo 1 se procederá a manipular el paquete si se activa la variable TRABAJO_ESTACION_1, para que estado suceda las variables PT08_MOVE_F_A y M102_F_A deben ser TRUE. La variable PT08_MOVE_F avisa que se aproxima un paquete, y la variable M102_F_A indica que no se trata de un paquete vacío.

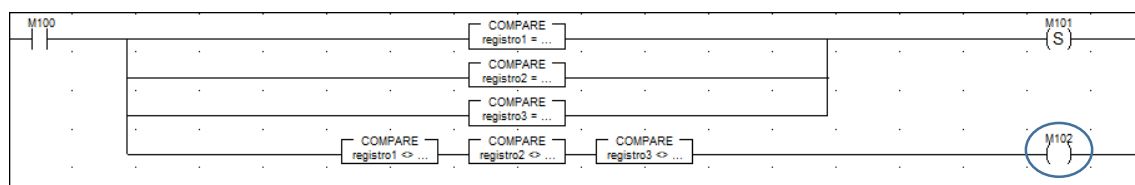


Figura 105 - Línea Ladder activación (M102)

Una vez la variable TRABAJO_ESTACION_1 este activa no se procederá a realizar la acción hasta que el paquete este en la posición de manipulación. Los temporizadores simularan los tiempos de procesado.

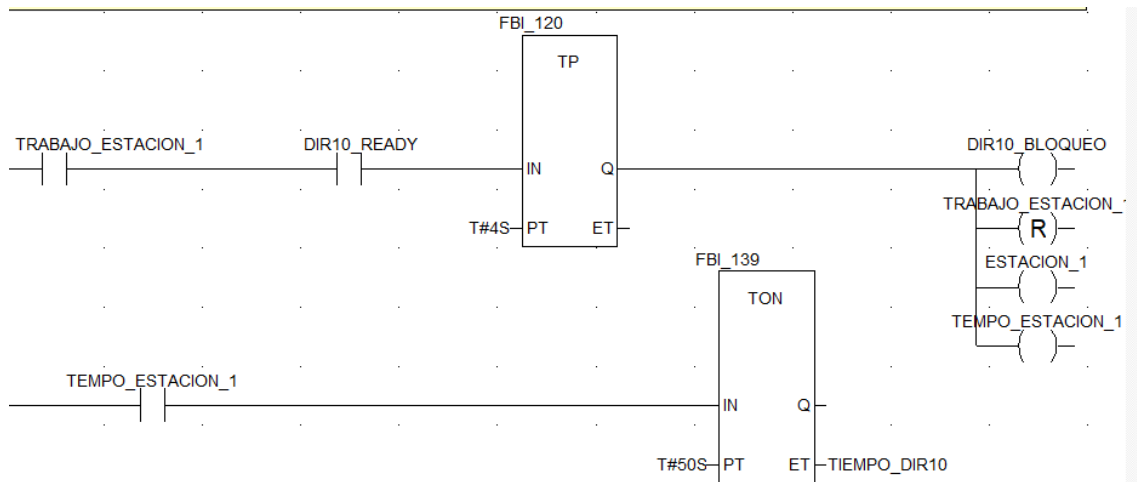


Figura 106 - Línea Ladder activación y temporización (estación 1)

Se guardará los tiempos de procesado en una variable y se incrementara un contador para informarnos del número de paquetes que han sido tratados en la estación. Una vez finalizada la acción se procederá a guardar en una ARRAY el tiempo que ha estado el producto en la estación y el ID correspondiente al paquete.

```
IF TEMPO_ESTACION_1 = TRUE THEN
    TIEMPO_E_1 := TIEMPO_DIR10;
    FLANCO_D_E_1 := FALSE;
END_IF;
```

Figura 107 - Línea ST (TIEMPO_E_1)

```
IF (TEMPO_ESTACION_1_F_A) = TRUE THEN
    INC (CONTADOR_TIEMPO_E_1);
    IF CONTADOR_TIEMPO_E_1 = 5 THEN
        ARRAY_TIEMPO_E_1 := ARRAY_TIEMPO_VACIA;
        CONTADOR_TIEMPO_E_1 := 1;
    END_IF;
    ARRAY_TIEMPO_E_1[CONTADOR_TIEMPO_E_1].CRONC := TIEMPO_E_1;
    ARRAY_TIEMPO_E_1[CONTADOR_TIEMPO_E_1].ID := escrituraRFID02;
    FLANCO_D_E_1 := TRUE;
END_IF;
```

Incrementar contador

Guardar tiempo e ID del paquete

Figura 108 - Línea ST (Estación de trabajo 1)

Después de haber pasado por todas las estaciones de trabajo que le correspondían, el paquete se dirigirá a la estación de extracción. El primer paso es saber si se trata de un paquete vacío o de un paquete finalizado.

Se procederá a extraer si la variable BANDEJA_CON_PRODUCTO se activa. Para ello la variable PT04_MOVE_F_A (indica que el paquete está situado en la posición de extracción) y la variable VAR_PRODUCTO_FIN (se trata de paquete finalizado) deben ser TRUE.

```
DIR04_READY_E := DIR04_READY;
DIR04_READY_F_A := RE (DIR04_READY_E);

IF PT04_MOVE_F_A = TRUE AND VAR_PRODUCTO_FIN = TRUE THEN

    BANDEJA_CON_PRODUCTO := TRUE; (* PONE A TRUE PARA INFORMAR QUE EL PAQUETE YA HA FINALIZADO*)
    VAR_PRODUCTO_FIN := FALSE;
```

Figura 109 - Línea ST activación (BANDEJA_CON_PRODUCTO)

La variable VAR_PRODUCTO_FIN se activará después de verificar que no se trata de un producto vacío, se ha programado el siguiente código para diferenciar los paquetes vacíos de los paquetes finalizados.

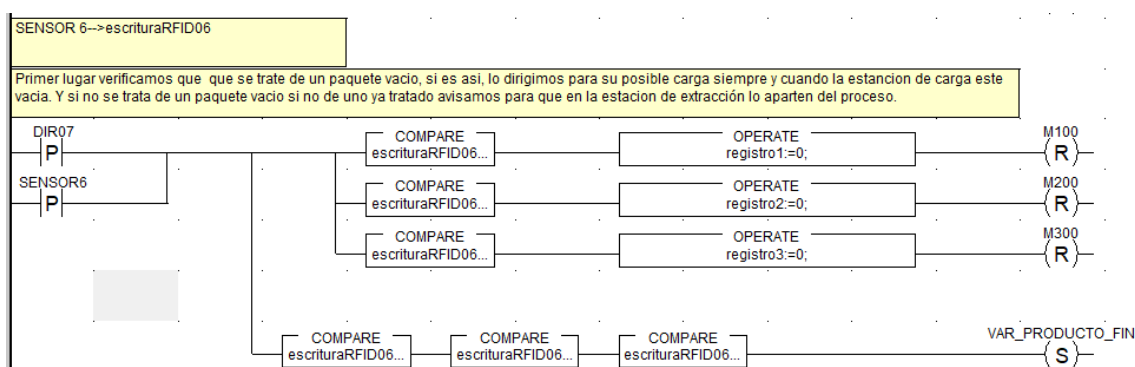


Figura 110 - Línea Ladder activación (VAR_PRODUCTO_FIN)

Una vez se estén las dos variables activadas se procera a simular el proceso con los temporizadores. Como en las estaciones anteriores se guardará en una ARRAY el tiempo de actividad junto al ID del paquete.

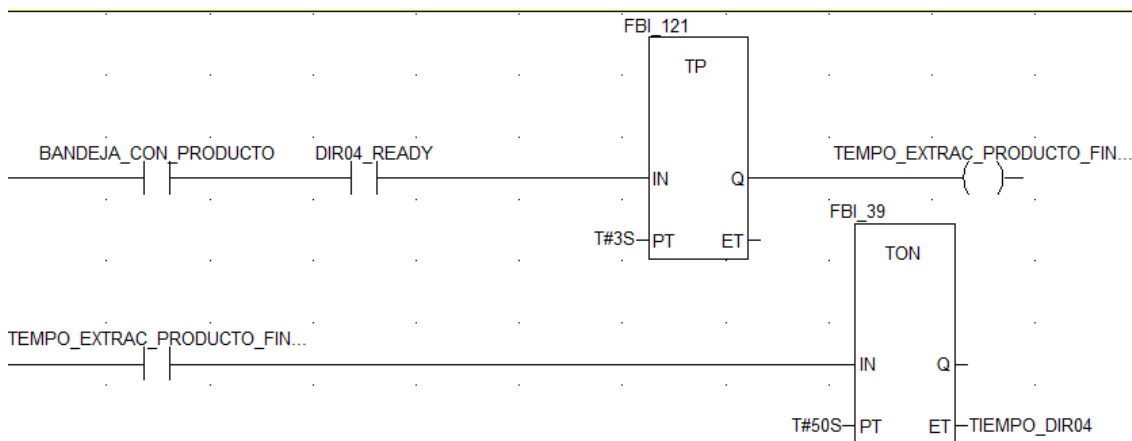


Figura 111 - Línea Ladder activación y temporización (estación de extracción)

6 Validación

Durante la programación de la celda industrial se han realizado una serie de pruebas para verificar el funcionamiento de cada una de las variables de nuestro sistema. Posteriormente se ha comprobado que las secciones de nuestro programa ejecutan perfectamente lo deseado y finalmente la funcionalidad del todo el programa.

Se ha dividido la validación en dos ensayos, pruebas funcionales y pruebas operacionales. Las pruebas funcionales se han realizado para comprobar pequeñas partes del programa o bloques individuales, no se realizan desde el punto de vista del automatismo sino observando que realizan bien la función de manera independiente. Las pruebas operacionales se han realizado para verificar un conjunto (bastantes líneas de código, secciones, programa entero), y se han realizado desde el punto de vista del automatismo.

Para llevar a cabo las pruebas se han elaborado las Pantallas del operador y Tablas de animación que permiten interaccionar con la variable y simular el proceso.

El proceso de validación se ha llevado a cabo con todas las variables y secciones del programa. A continuación, se citará algunos ejemplos representativos del proceso.

Variable del retenedor

La prueba consistía en observar si se activaba la variable direccionada al retenedor (DIR03_down), y tal y como se esperaba la variable se activó.

Variable de la plataforma

La prueba se basaba en validar la activación de la variable de subida de la plataforma (SUBE_PT05) que permite la circulación entre líneas. Se esperaba que una vez activada la variable PT05_MOVE_DIR08 se pusiera a 1 (SUBE_PT05). El resultado fue el esperado.

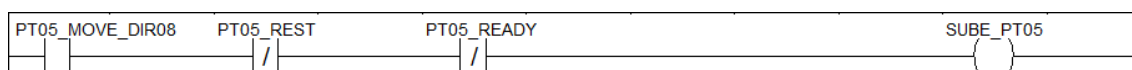


Figura 112 - Línea Ladder activación (SUBE_PT05)

Variable del motor

La prueba consistía en observar si se activa la variable de salida del motor una vez activa la de avance (siempre y cuando se cumplan las otras condiciones). El resultado fue el esperado.

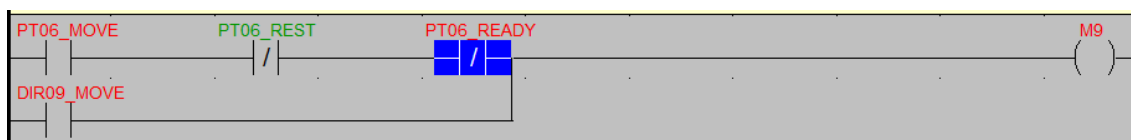


Figura 113 - Línea Ladder activación (M9)

Lectura del Identificador

El ensaño consiste en realizar la lectura del paquete y observar que se guarda el valor en la variable. Se ha simulado y se ha obtenido el resultado esperado. Los dígitos introducidos se han guardado correctamente.

INTRODUCCIÓN Y ACTIVACIÓN

SENSOR 1

1234

SENSOR 4

2345

RFID 1

RFID 4

ID SENSORES

1234

RFID 01

2345

RFID 04

Figura 114 - Pantalla del operador (RFID01, RFID04)

Selección de un dígito

La prueba consiste seleccionar el dígito de interés una vez leído el identificador. El resultado no era el esperado en un principio ya que solo funciona si se trataba del ultimo dígito. Para solucionar el problema se trata el dígito en formato BCD y se ha utilizado la función de desplazar bits.

escrituraRFID01	1234	INT
%MW1	4	INT

Figura 115 - Ejemplo Tabla animación (escrituraRFID01)

Estaciones de trabajo

La prueba consistía en observar si una vez cumplidas las condiciones se activarían los temporizadores que simulan el trabajo de las estaciones. El resultado fue el esperado.

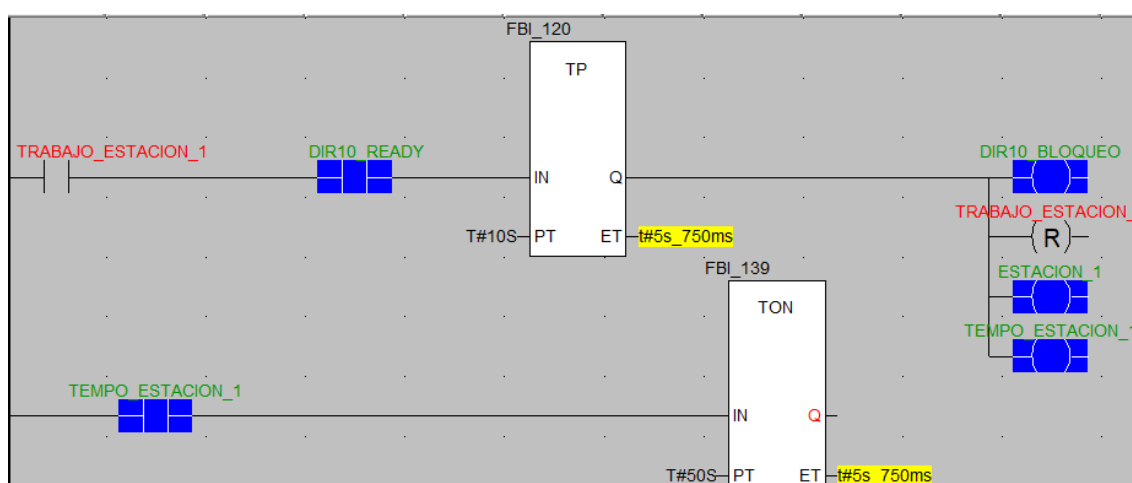


Figura 116 - Línea Ladder simulación (Estación de trabajo)

Temporizador y visualización estación de trabajo

La prueba consistía en comprobar que una vez activada la estación de trabajo, temporizaba correctamente, además de guardar el tiempo de procesado y visualizar el tiempo real de ejecución. El resultado fue el esperado.

●	TIEMPO E_M_P	0s	TIME	Tiempo estación Materias Primas
●	TIEMPO E_P_F	19s_975ms	TIME	Tiempo estación Extraccion Producto Final

Figura 118 - Tabla animación (tiempo de trabajo)

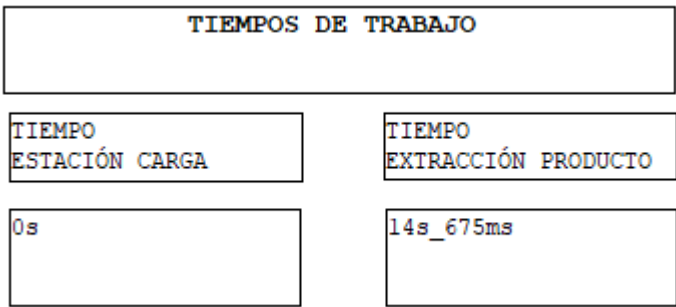


Figura 118 - Pantallas de operador (visualización tiempo real)

Sección RFID 1 (selección)

La prueba consistía en observar si se activaba la selección del producto una vez leído el identificador del paquete. El resultado no fue el esperado ya que a veces se activaban dos productos a la vez. La modificación para solucionar el error fue arreglar los rangos.



Figura 119 - Pantallas de operador introducción (RFID01)

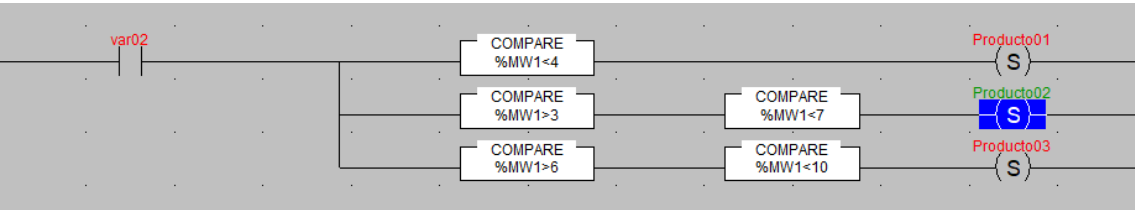


Figura 120 - Línea Ladder activación (Producto2)

Sección RFID 2,3,5 (bifurcación)

La prueba consistía en verificar si se activa la dirección correcta una vez leído el identificador del paquete. El resultado fue el esperado.

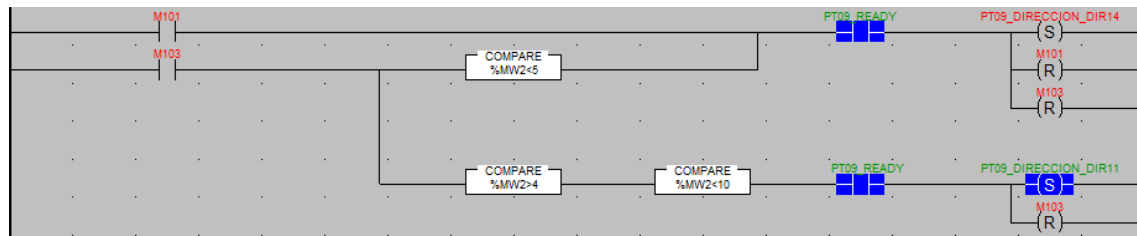


Figura 121 - Línea Ladder activación dirección (PT09_DIRECCION_DIR11)

Sección RFID 4,6 (registros)

El ensayo se basaba en realizar la lectura de un paquete en el sensor 4 y observar si se registraba el identificar en la variable (registro1, registro2, registro3). Posteriormente realizar la lectura en el sensor 6 del mismo identificador y observar que se borrara de las variables (registro1, registro2, registro3). El resultado fue el esperado en cuanto a eliminar el identificador de los registros, en cambio no fue el esperado en el guardado. La acción correctora fue añadir temporizadores (TON).

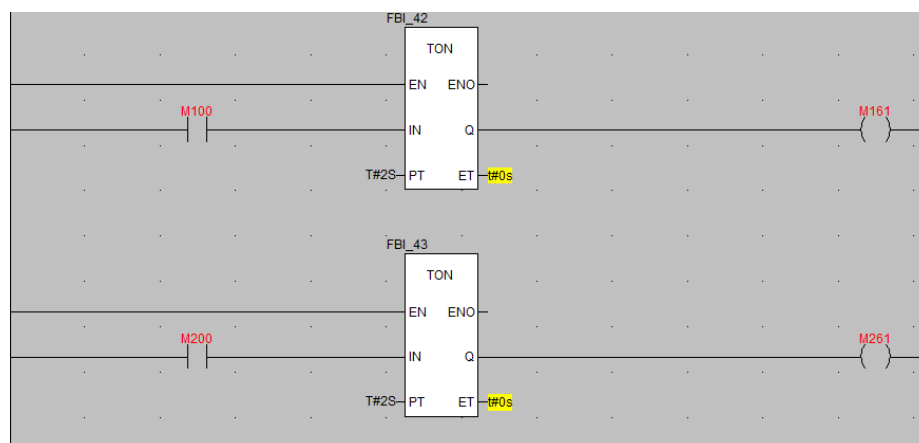


Figura 122 - Línea Ladder acción correctora (Temporizador TON)

Comprobación sección de visualización a tiempo real

La prueba consistía en observar que todas las variables de visualización se observaban a tiempo real. El resultado fue el esperador.

ID SENSORES		TIEMPOS DE TRABAJO	
1234	RFID 01	TIEMPO ESTACIÓN CARGA	TIEMPO EXTRACCIÓN PRODUCTO
0	RFID 04	0s	19s_975ms
0	RFID 05		

Figura 123 - Pantallas de operador (ID sensores, Tiempos de trabajo)

Comprobación sección de trazabilidad

La prueba de la trazabilidad consistía en poner en marcha una estación de trabajo y observar si guardaba su ID y el tiempo de procesado del paquete. El resultado ha sido el esperado.

ID SENSORES	
1500	RFID 01
0	RFID 04
0	RFID 05
4567	RFID 06

Figura 125 - Pantallas operador (ID sensores)

ARRAY_TIEMPO_P_F		ARRAY[1..4] OF...
ARRAY_TIEMPO_P...		TIPO_TIEMPO
CRONO	19s_975ms	TIME
ID	4567	INT
ARRAY_TIEMPO_P...		TIPO_TIEMPO
ARRAY_TIEMPO_P...		TIPO_TIEMPO
ARRAY_TIEMPO_P...		TIPO_TIEMPO
ARRAY_TIEMPO_M_P		ARRAY[1..4] OF...
ARRAY_TIEMPO_M...		TIPO_TIEMPO
CRONO	1s_975ms	TIME
ID	1500	INT

Figura 125 - Tabla de animación (ARRAY_TIEMPO_P_F/ARRAY_TIEMPO_M_P)

Se ha elaborado una serie de videos sobre la simulación de alguno del proceso del proyecto, en el documento **ANNEX_SIMULACIÓN** se explica brevemente el contenido que se trata en el video.

7 Conclusiones

Partiendo de una celda industrial ubicada en el la Universidad Politécnica de Cataluña se ha realizado la simulación del proceso y se ha logrado cumplir con éxito todos los objetivos establecido ya que hemos podido validar todo el proceso del proyecto.

El objetivo principal del proyecto consiste en gestionar la distribución de los paquetes mediante una aplicación industrial basada en la automatización. Para logra dicho objetivo ha sido necesario comprender la celda de trabajo donde iba a realizar la simulación, las comunicaciones entre los PLCs de la celda, entender el funcionamiento de cada elemento y finalmente programa los PLCs. Esto me ha ayudado a comprender el funcionamiento de una celda industrial real, entender las comunicaciones entre PLCs y reforzar mis conocimientos en la programación.

Otro objetivo consistía en seleccionar la mejor tecnología para identificar los paquetes. Analizar la celda y decidir que tecnología de identificación disponibles en el mercado se ajustaba mejor a nuestras necesidades y a las de la celda industrial. Esto me ayudado a comprender la necesidad de realizar un estudio antes de adquirir cualquier producto. Comparar las diferentes opciones que existen y posteriormente observar que se pueda adaptar a tu entorno de trabajo.

Después de realizar el proyecto industrial he comprendido la dificultad de gestionar una celda de trabajo. He entendido la importancia de analizar cualquier aspecto, fijarte en la compatibilidad de los distintos elementos a instalar y en realizar un código de programación estructurado y eficiente.

El proyecto me ha ayudado a reforzar mis conocimientos en la automatización, entender la comunicación entre distintos PLCs, reforzar mis habilidades en la programación, aprender a seleccionar la opción que mejor se adapte a tu entorno de trabajo, comprender la importancia de los RFID en la industria 4.0 y aprender conceptos de los protocolos de comunicación.

En cuanto a una posible continuidad del proyecto en el futuro, a continuación, numeraré varias recomendaciones que serían interesantes de añadir.

1. Controlar y supervisar el proceso industrial con el sistema SCADA.
2. Añadir un editor de flujo basado en el navegador donde se puede añadir o eliminar nodos y conectarlos entre sí, esto permite conectar los dispositivos de hardware, APIs y servicios en líneas. Node-RED es una herramienta de programación visual que cumple estas funciones.
3. Añadir una base de datos para gestionar todos los datos que genera la línea. Y mediante algún tipo de software visualizar los datos.
4. Añadir una estación de control de calidad, para controlar la calidad de nuestros productos.
5. Un panel o algún sistema de control que nos informara de los errores de producción y averías de los elementos de la línea.
6. Llevar a cabo la instalación de los sensores RFID en la celda de trabajo.

8 Bibliografía

1. IEC Functional Safety and IEC 61508. <https://www.iec.ch/functionalsafety/>. Accessed June 28, 2020.
2. PROFIBUS: Qué es, cómo funciona y tipos | Comunicaciones Industriales. <https://www.cursosaula21.com/que-es-profibus/>. Accessed June 16, 2020.
3. Introducción a CAN - National Instruments. <https://www.ni.com/es-es/innovations/white-papers/06/controller-area-network--can--overview.html#section-1250675107>. Accessed June 16, 2020.
4. Diego Romero IM. *Introducción a Ethernet Industrial Curso Intensivo AAECA*.; 2007.
5. Pro MU. Manual Unity Pro.
6. Dispositivo de parada de emergencia ES... | EUCHNER – More than safety. <https://www.euchner.de/es-es/productos/dispositivos-de-parada-de-emergencia/dispositivo-de-parada-de-emergencia-es/>. Accessed June 1, 2020.
7. Ferrer JM. Tecnología RFID, habilitador necesario de la Industria 4.0. http://www.automataeinstrumentacion.com/es/notices/2020/06/tecnologia-rfid-habilitador-necesario-de-la-industria-4.0-46733.php?utm_source=newsletter&utm_medium=email&utm_campaign=tecnologia-rfid-habilitador-necesario-de-la-industria-4.0#.XvNolmgzbIV. Accessed June 24, 2020.
8. Barras C De. RFID vs. Código de Barras.
9. María J, Herrera C, Casanovas ES. *ESTUDIO, DISEÑO Y SIMULACIÓN DE UN SISTEMA DE RFID BASADO EN EPC*.

PLC

¿Qué es y para qué sirve un PLC? - Ingeniería Mecafenix.

<https://www.ingmecafenix.com/automatizacion/que-es-un-plc/>. Accessed June 29, 2020.

Lenguajes para programación de plc - Ingeniería Mecafenix.

<https://www.ingmecafenix.com/automatizacion/lenguajes-programacion-plc/>. Accessed June 29, 2020.

PAC-Performance-Centered Adaptive Curriculum for Employment Needs.

Grafcet. <https://www.automatas.org/redes/grafcet.htm>. Accessed June 29, 2020.

Webmaster. Mundo de los Automatas y sus accesorios.

Protocolos de comunicación

Rodríguez-Aragón LJ, Rey U, Carlos J. *Tema 4: Internet y Teleinformática Tema 4: Internet y Teleinformática Informática Básica*.

Características de un sistema CAN Bus en el automóvil. <https://petrolheadgarage.com/cursos-automocion/caracteristicas-de-un-sistema-can-bus/>. Accessed June 29, 2020.

PROFIBUS. <http://www.etitudela.com/celula/downloads/2profibus.pdf>. Accessed June 29, 2020.

Protocolo Ethernet: Características y Funcionamiento » CCNA 200-301.

<https://ccnadesdecero.es/protocolo-ethernet-caracteristicas/>. Accessed June 29, 2020.

Qué es el protocolo Ethernet Industrial y cómo funciona | Aula21.

<https://www.cursosaula21.com/que-es-ethernet-industrial/>. Accessed June 29, 2020.

Elemetos de la celda

¿Qué es un sensor fotoeléctrico? | Fundamentos del sensor: Guía de sensores para fábricas clasificados por principios | KEYENCE.

<https://www.keyence.com.mx/ss/products/sensor/sensorbasics/photoelectric/info/>. Accessed June 29, 2020.

Sensor de proximidad capacitivo - Ingeniería Mecafenix.

<https://www.ingmecafenix.com/automatizacion/sensor-proximidad-capacitivo/>. Accessed June 29, 2020.

Pulsador. <https://www.areatecnologia.com/electricidad/pulsador.html>. Accessed June 29, 2020.

Tecnología de identificación

Barras C De. RFID vs. Código de Barras.

RFID - Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/RFID>. Accessed June 29, 2020.

Serna Mulero R. Implantación de un sistema RFID para obtener trazabilidad en la cadena de suministros. 2012:1 recurs electrònic. <http://hdl.handle.net/2099.1/15545>.

María J, Herrera C, Casanovas ES. *ESTUDIO, DISEÑO Y SIMULACIÓN DE UN SISTEMA DE RFID BASADO EN EPC*.